



TRABAJO FIN DE GRADO
GRADO DE INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES
DESARROLLO DEL SISTEMA DE CONTROL DE CONVERTIDORES
DE POTENCIA CONMUTADOS CC/CA.

VICTOR ARNAIZ GARCIA-MORA
TUTOR: PABLO ZUMEL VAQUERO

INDICE

CAPITULO 1.	9
INTRODUCCION	9
1.1- Explicación del trabajo y sus objetivos.	9
1.2-Estructura del documento.	9
1.3-Herramientas empleadas.....	10
1.4- Marco Regulador.....	11
CAPITULO 2.	12
INTRODUCCIÓN AL CONVERTIDOR INVERSOR DE PUENTE COMPLETO	12
2.1 Introducción al convertidor inversor de puente completo.....	12
2.2 Aplicaciones del convertidor inversor de puente completo.	13
2.3 Modulación del convertidor inversor Full Bridge.....	15
2.3.1 Modulación por desplazamiento de fase.....	15
2.3.2 Modulación PWM Bipolar.	16
2.3.3 Contraste entre modulaciones.....	18
2.4 Análisis estático	19
2.4.1 Análisis estático en modulación por desplazamiento de fase	19
2.4.2 Análisis estático en modulación por PWM bipolar	23
2.5 Control del convertidor inversor puente completo	25
2.5.1 Control en lazo Abierto	26
2.5.2 Control en lazo Cerrado	26
CAPITULO 3.	28
PROTOTIPO CONVERTIDOR INVERSOR PUENTE COMPLETO	28
3.1 Puente de transistores	28
3.2 Tiva C Series EK-TM4C1294XL Texas Instruments.....	29
3.3 Adaptación de la tensión de salida	30
3.3.1 Divisor resistivo	30
3.3.2 Eliminar tensiones negativas.....	32
3.4 Componentes para el montaje del circuito.....	34
3.5 Alimentación.	35
3.6 Conexión en modulación por desplazamiento de fase.....	36
CAPITULO 4	38
PROGRAMACIÓN DEL SISTEMA.....	38
4.1 Funcionamiento y decisiones de diseño.	38

4.2 Recursos empleados	39
4.2.1 Señales PWM.....	39
4.2.2 Timer	40
4.2.3 ADC.....	40
4.2.4 UART (Comunicación puerto serie)	40
4.3 Código.....	41
4.3.1 Variables globales en modulación por desplazamiento de fase	41
4.3.2 Variables globales en modulación por desplazamiento de fase	42
4.3.3 Set Up	43
4.3.4 Diagramas de flujo y explicación de las funciones principales.....	52
CAPITULO 5	60
PRUEBAS EXPERIMENTALES	60
5.1 Modulación por desplazamiento de fase	60
5.1.1 Lazo Abierto	60
5.1.2 Lazo Cerrado.....	61
5.2 Modulación PWM Bipolar	62
5.2.1 Lazo Abierto	62
5.2.2 Lazo Cerrado.....	63
CAPITULO 6	64
EJERCICIO PROPUESTO A ALUMNOS DE LA ASIGNATURA ELECTRÓNICA DE POTENCIA	64
6.1 Cálculos teóricos previos.....	64
6.1.1 Cálculos para modulación por desplazamiento de fase.....	64
6.1.2 Cálculos para modulación por PWM bipolar.....	65
6.2 Trabajo en el laboratorio.....	67
6.2.1 Modo desplazamiento de fase	67
6.2.2 Modo PWM bipolar	69
CAPITULO 7	71
PRESUPUESTO Y PLANIFICACIÓN SEMANAL	71
7.1 Presupuesto y costes de amortización.....	71
7.2 Planificación semanal.....	74
CAPITULO 8	76
CONCLUSIÓN Y TRABAJOS FUTUROS	76
BIBLIOGRAFÍA:.....	78
ANEXOS	79

A. CÓDIGO DE PROGRAMA PARA MODULACIÓN POR DESPLAZAMIENTO DE FASE:.....	79
B. CÓDIGO DE PROGRAMA PARA MODULACIÓN PWM BIPOLAR.....	86

INDICE DE FIGURAS

Figura 1. Esquema del convertidor inversor de puente completo.....	13
Figura 2. Esquema de control de velocidad en motor.....	14
Figura 3. Esquema de adaptación de tensión generada por aerogenerador.....	14
Figura 4. Esquema de adaptación de tensión generada por panel solar.....	15
Figura 5. Circuito del convertidor inversor.....	15
Figura 6. Señales de gobierno de los transistores en modulación por desp de fase.....	16
Figura 7. Variación de tensión media por variación de ancho de pulso.....	17
Figura 8. Tensión de salida V_{AB} en modulación PWM Bipolar.....	18
Figura 9. Contraste V_O para capacidades de 3.3 uF vs 10 uF en desp de fase PSIM.....	19
Figura 10. Tensión de salida V_{AB} en modulación por desp de fase.....	20
Figura 11. Contraste V_O para capacidades de 3.3 uF vs 10 uF en desp de fase REAL.....	23
Figura 12. Tabla de amplitudes normalizadas.....	24
Figura 13. V_O para modulación PWM Bipolar.....	25
Figura 14. Lazo Abierto.....	26
Figura 15. Lazo Cerrado.....	27
Figura 16. Puente Inversor.....	28
Figura 17. Esquema del puente inversor en PSIM.....	28
Figura 18. Tiva C Series EK-TM4C1294XL	30
Figura 19. Divisor resistivo en PSIM.....	31
Figura 20. Tensión de salida V_O vs tensión después de divisor resistivo.....	31
Figura 21. AD620.....	32

Figura 22. Esquema de Ajuste de Offset.....	33
Figura 23. V_{Out} del integrado AD620.....	34
Figura 24. Imagen de conexionado de fuentes de alimentación.....	35
Figura 25. Esquema de conexionado para modulación por desp de fase.....	36
Figura 26. Circuito Real.....	37
Figura 27. Estrategia de generación de PWM en modulación por desp de fase.....	45
Figura 28. Señales de gobierno de los transistores en modulación por desp de fase.....	47
Figura 29. Tensión V_A y V_B teóricas.....	48
Figura 30. Tensión V_A y V_B reales.....	48
Figura 31. Gráfico de valores de la variable $\sin[20]$	49
Figura 32. Tiempos muertos.....	49
Figura 33. Flujograma de la función Main (programa principal).....	53
Figura 34. Menú principal.....	53
Figura 35. Flujograma de la función MenuPrincipal().....	54
Figura 36. Menú de selección de desfase.....	55
Figura 37. Flujograma de la función MenuDesfase().....	55
Figura 38. Flujograma de la función LazoAbierto().....	56
Figura 39. Menú modificar Deadband.....	56
Figura 40. Flujograma de la función ModificarDeadBand().....	57
Figura 41. Flujograma de la función LazoCerrado().....	58
Figura 42. Flujograma de la función ModificarAmplitud(seleccion).....	59
Figura 43. Amplitud vs Desfase.....	61
Figura 44. Amplitud vs modulación en amplitud (A).....	63

INDICE DE TABLAS

Tabla 1. Valor de variables del circuito.....	21
Tabla 2. Armónicos en desp. de fase para $C=10\text{ uF}$	21
Tabla 3. Armónicos en desp. de fase para $C=3.3\text{ uF}$	22
Tabla 4. Armónicos en PWM Bipolar para $C=3.3\text{ uF}$	25
Tabla 5. Recursos empleados del micro Tiva.....	39
Tabla 6. Variables en modulación por desp. de fase	41
Tabla 7. Variables en modulación PWM Bipolar.....	42
Tabla 8. Datos experimentales en lazo abierto para desp. de fase.....	60
Tabla 9. Datos experimentales en lazo cerrado para desp. de fase.....	61
Tabla 10. Datos experimentales en lazo cerrado para desp. de fase.....	61
Tabla 11. Datos experimentales en lazo abierto para PWM Bipolar.....	62
Tabla 12. Datos experimentales en lazo cerrado para PWM Bipolar.....	63
Tabla 13. Datos experimentales en lazo cerrado para PWM Bipolar.....	63
Tabla 14. Trabajo del alumno en prácticas.....	64
Tabla 15. Trabajo del alumno en prácticas.....	65
Tabla 16. Trabajo del alumno en prácticas.....	65
Tabla 17. Trabajo del alumno en prácticas.....	66
Tabla 18. Trabajo del alumno en prácticas.....	66
Tabla 19. Tiempo de uso de equipos y software.....	71
Tabla 20. Coste de materiales.....	72
Tabla 21. Amortización de equipos.....	72
Tabla 22. Amortización de licencias.....	72
Tabla 23. Recursos humanos.....	72
Tabla 24. Planificación semanal.....	74

CAPITULO 1.

INTRODUCCION

1.1- Explicación del trabajo y sus objetivos.

El principal objetivo del trabajo consiste en la elaboración de un sistema de control para un convertidor inversor de puente completo, para su posterior implementación en las prácticas de la asignatura electrónica de potencia de forma que, una vez desarrollado, sirva con fines didácticos a alumnos de distintas asignaturas.

Para ello, en primer lugar se llevará a cabo un estudio teórico del sistema, hallando determinados valores que condicionan el comportamiento del inversor, además de una propuesta de control. Una vez realizado el estudio teórico, se pasará a desarrollar un prototipo para su posterior uso en prácticas de la asignatura previamente nombrada.

Además de lo descrito anteriormente, se llevará a cabo el desarrollo de una interfaz entre el usuario y el convertidor, a través del microcontrolador Tiva C Series EK-TM4C1294XL, la aplicación será intuitiva y permitirá al usuario modificar ciertos valores y modos de funcionamiento del convertidor para ver de esta manera como responde el convertidor a estos cambios en tiempo real.

Por lo tanto los objetivos parciales del trabajo son:

1. Estudio teórico de la planta del convertidor inversor.
2. Desarrollo de un sistema de control.
3. Montaje del prototipo a utilizar en prácticas.
4. Desarrollo de la aplicación que servirá de interfaz entre el convertidor y el usuario.

1.2-Estructura del documento.

A continuación se hará un breve resumen del contenido de cada capítulo.

En el **capítulo 1** se hará una breve introducción en la cual se presentan los objetivos del proyecto, la estructura del documento y las herramientas empleadas.

En el **capítulo 2** se introducirá la clasificación general de tipos de convertidores, se explicarán algunas de las aplicaciones del convertidor inversor analizado en el presente trabajo, las modulaciones que van a ser utilizadas y un pequeño contraste entre ambas. Por otro lado se llevará a cabo un análisis teórico del convertidor para cada tipo de modulación y se explicará brevemente en qué consiste el control en lazo abierto y lazo cerrado.

En el **capítulo 3** se presentarán los componentes que han sido necesarios para el montaje, también hablaré de cada una de las etapas necesarias para acondicionar la señal de salida del inversor y poder ser leída por el ADC de la placa e incluso sobre el tipo de alimentación y los valores de tensión requeridos.

En el **capítulo 4** el tema principal a tratar es la programación del micro controlador Tiva C Series EK-TM4C1294XL. Se explicarán cada una de las variables utilizadas así como las funciones necesarias para habilitar los distintos periféricos empleados (ADC, UART, Timers...). Por otro lado se introducirá la estrategia de generación que se ha llevado a cabo para los dos tipos de modulación. Finalmente, se explicarán las funciones que aparecen en la programación y determinados algoritmos empleados así como sus diagramas de flujo para facilitar su comprensión.

En el **capítulo 5** se tomarán una serie de medidas experimentales, tanto en lazo abierto como en lazo cerrado para comprobar que el funcionamiento del sistema es el esperado.

En el **capítulo 6** se proponen una serie de ejercicios para llevar a cabo en las prácticas de la asignatura Electrónica de Potencia y de esta forma facilitar la comprensión de este tipo de convertidores.

En el **capítulo 7** se presenta el presupuesto aproximado para la realización del presente proyecto.

Finalmente, en el capítulo 8 se comentarán las conclusiones a las cuales se ha llegado tras la realización del trabajo y algunos trabajos futuros que podrían enlazarse con este proyecto.

1.3-Herramientas empleadas.

Para el desarrollo del presente trabajo, ha sido necesario el uso de numerosas herramientas de análisis, simulación y desarrollo de software que se citan a continuación:

1. PSIM: Mediante PSIM se han planteado los circuitos que gobiernan el sistema real, y se ha podido simular su respuesta antes de llevar a cabo el montaje del sistema físico.

2. TERA TERM: Este programa es el entorno en el cual el usuario podrá leer un menú y podrá seleccionar distintos parámetros a modificar en la planta. Además, su función principal es la de establecer una conexión por protocolo serie (UART) entre el microcontrolador y el PC.

3. CODE COMPOSER STUDIO: Es un entorno de desarrollo de sistemas integrados y de sistemas empotrados de Texas Instruments. En él, podremos programar y coordinar los distintos recursos que posee el microcontrolador Tiva C series EK-TM4C1294XL para llevar a cabo el control del sistema.

Además de los recursos empleados, es necesario citar aquellas asignaturas del grado en ingeniería en Tecnologías Industriales cuyos conocimientos adquiridos a lo largo del curso, han servido para poder desarrollar el presente trabajo, además de reforzar y adquirir nuevos conocimientos, y las destrezas adquiridas en cada una de ellas:

- **Fundamentos de ingeniería Electrónica:** En esta asignatura se aprende sobre el funcionamiento de los distintos componentes electrónicos, así como el funcionamiento del transistor MOSFET cuyo uso en el presente trabajo es imprescindible.
- **Instrumentación Electrónica:** Capacidad para analizar la respuesta en frecuencia de los distintos sistemas electrónicos, la conversión analógico-digital, y el tratamiento de señales.
- **Fundamentos de ingeniería Eléctrica:** Capacidad para analizar circuitos mediante diferentes métodos.
- **Ingeniería de Control I:** Capacidad para analizar sistemas continuos y diseñar distintos reguladores para dichos sistemas.
- **Ingeniería de Control II:** Capacidad para analizar sistemas discretos y diseñar reguladores para sistemas dinámicos lineales.
- **Electrónica de Potencia:** Capacidad para reconocer las distintas topologías de convertidores de potencia (convertidores CC/CC, convertidores CC/CA, convertidores CA/CC).
- **Electrónica digital y sistemas Electrónicos Digitales:** Capacidad para entender el funcionamiento de un micro controlador, además de poder analizar y extraer información del datasheet sobre los recursos de los que dispone el micro, pines a utilizar, velocidades de conversión, ADC, timers, comunicación series etc.

1.4- Marco Regulador.

Las normas que se siguieron y las restricciones y prevenciones que se acataron a la hora de desarrollar el proyecto fueron las descritas en la normativa de uso de equipos de laboratorio de la Universidad Carlos III de Madrid y en el documento de prevención de riesgos laborales [1].

INTRODUCCIÓN AL CONVERTIDOR INVERSOR DE PUENTE COMPLETO

En este capítulo se introducirá el convertidor sobre el cual se trabajará a lo largo del presente trabajo fin de grado, así como los diferentes convertidores existentes. Se realizarán los cálculos teóricos correspondientes al convertidor inversor de puente completo para posteriormente compararlos con los datos obtenidos mediante la simulación en PSIM y las medidas sobre el sistema real.

2.1 Introducción al convertidor inversor de puente completo.

Un convertidor de potencia es un sistema electrónico cuya finalidad es la conversión de energía eléctrica de un formato a otro, por ejemplo obtener corriente alterna a partir de continua o viceversa.

Existen diversos criterios para clasificar convertidores, el más común es aquel que los clasifica según el formato de energía de entrada y de salida. Según dicho criterio se pueden diferenciar cuatro grandes grupos:

- **Convertidores CC/CC:** Este tipo de convertidor transforma un valor determinado de corriente continua a la entrada en uno distinto de salida, además pueden incluir aislamiento galvánico. Su uso se restringe a sistemas embarcados, en los cuales la distribución de energía se hace mediante corriente continua. También es utilizado como intermediario tras una rectificación no controlada, con el fin de controlar esta tensión y de esta forma aumentar el factor de potencia del rectificador. Se clasifican en tres grandes grupos atendiendo al tipo de modificación realizada sobre la tensión de entrada, ya sea aumentarla (elevador), reducirla (reductor) o ambas (reductor-elevador) [7].
- **Convertidores CA/CC o rectificadores:** Este tipo de convertidor transforma la corriente eléctrica alterna monofásica o trifásica en corriente continua. Desde el punto de vista de los accionamientos, presenta una importancia fundamental, ya que se utilizan de forma general en las máquinas siguientes: máquinas de corriente continua, máquinas síncronas y asíncronas, y motores especiales, como el motor paso a paso o sin escobillas [7].
- **Convertidor CA/CA o transformador:** Este tipo de convertidores se usa en el diseño de arrancadores suaves para reducir la intensidad demandada durante el arranque de los motores de inducción. Su función principal es la de modificar el valor eficaz de la tensión de entrada, conservando su frecuencia, aunque también puede conseguirse una frecuencia submúltiplo de la frecuencia de entrada, en cuyo caso reciben el nombre de ciclo convertidores [7].

- **Convertidor CC/CA o inversor:** En este caso, la corriente de entrada es continua y es transformada en corriente alterna. Además ofrece la posibilidad de controlar tanto la frecuencia como el valor eficaz de la tensión o intensidad de salida, lo que les hace fundamentales para el diseño de accionamientos de máquinas síncronas y asíncronas. Como la distribución de energía se realiza en corriente alterna, suelen ir asociados a un rectificador. Uno de sus usos más comunes es el de acoplar una energía variable a la red, por ejemplo la producida por un generador eólico, el cual se caracteriza por un motor primario de velocidad muy variable [7].

En el presente trabajo se analizará el funcionamiento de un convertidor inversor (CC/CA) de puente completo. La terminología puente completo, es debida a que el convertidor posee un puente de cuatro transistores, aunque existen otras configuraciones, como la de medio puente o la configuración Push-Pull que además posee aislamiento galvánico. El esquema eléctrico del convertidor a analizar aparece reflejado en la **Figura 1** original de [4].

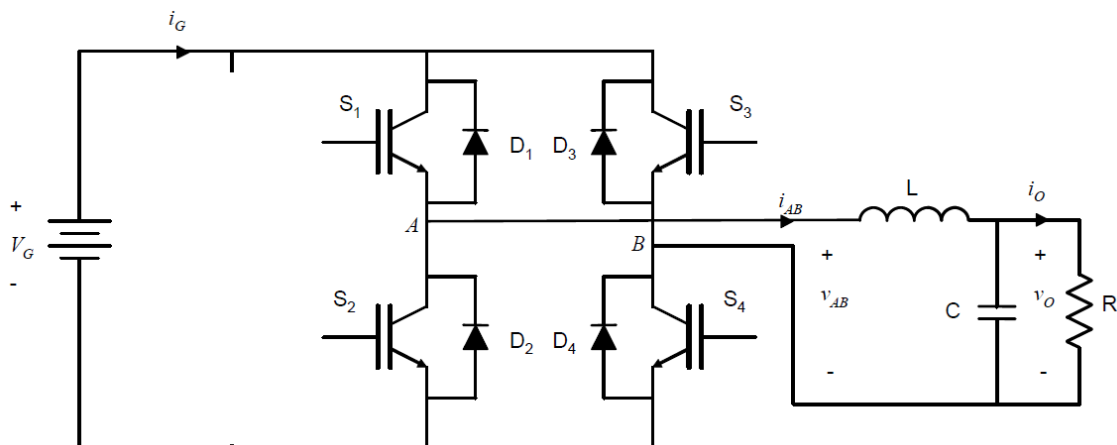


Figura 1 [4]

2.2 Aplicaciones del convertidor inversor de puente completo.

Su principal aplicación es regular la velocidad en motores de corriente alterna, tanto asíncronos como síncronos. En la **Figura 2** original de [4] se muestra el esquema eléctrico de control de motores trifásicos, para ello es necesario el uso de un puente inversor trifásico.

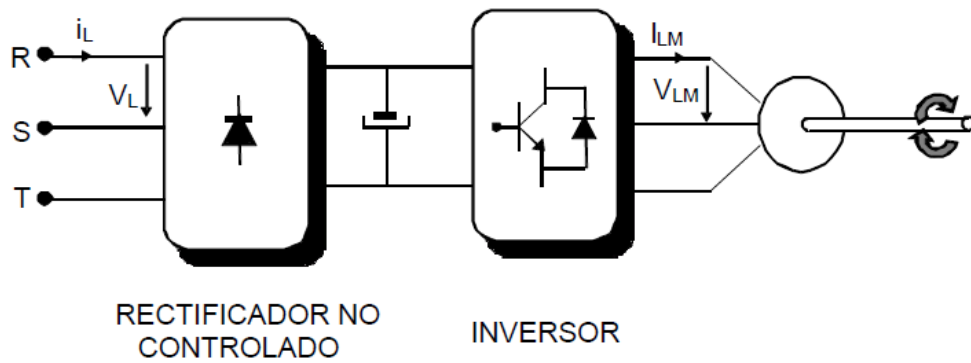


Figura 2 [4]

Variando el valor eficaz y la frecuencia de una tensión aplicada al estator de un motor de inducción, se logra variar la velocidad para diferentes pares. Hoy en día esta es una aplicación con un gran volumen de negocio, ya que el motor de inducción es muy sencillo, robusto, de mantenimiento casi nulo y actualmente el conjunto motor de inducción-variador de velocidad es muy competitivo en cuanto a su precio [4].

Algunas aplicaciones típicas de reguladores electrónicos de velocidad con puente inversor trifásico son:

- Tracción ferroviaria.
- Máquinas Herramientas.
- Sistemas de bombeo y ventilación.
- Cintas Transportadoras.
- Laminadoras y bobinadoras...

Otro de sus usos más destacable es con fines de acondicionamiento de la tensión generada por una estación aerogenerador.

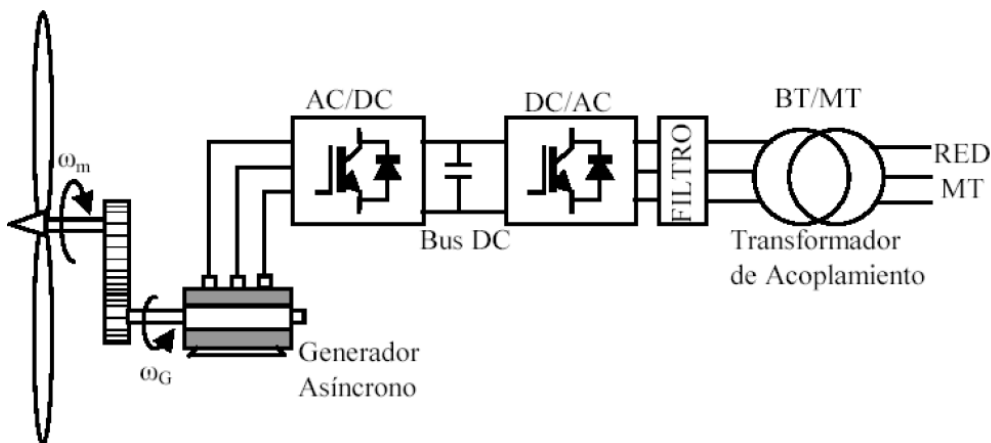


Figura 3 [4]

También es usado para el acondicionamiento de potencia para conexión a red de paneles solares fotovoltaicos, o para generar calentamiento por inducción. Para esta aplicación se emplea el puente inversor monofásico, el cual es objeto de estudio en el presente trabajo.

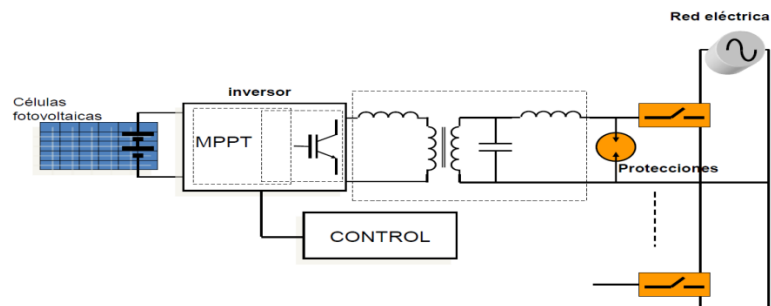


Figura 4 [4]

2.3 Modulación del convertidor inversor Full Bridge

En este apartado se explicarán los dos tipos de modulación que van a ser utilizadas. Por un lado se encuentra la modulación por desplazamiento de fase, y por otro la modulación por PWM Bipolar.

2.3.1 Modulación por desplazamiento de fase

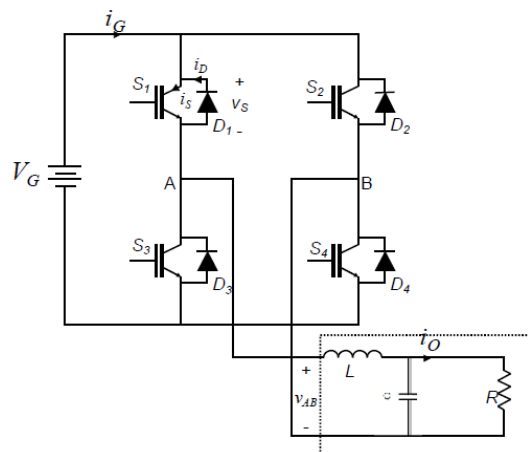


Figura 5 [4]

Las señales S_1 - S_3 y S_2 - S_4 son complementarias entre ellas, pero en esta modulación, en las señales de gobierno S_1 y S_2 , se introduce un desfase de manera que obtenemos el siguiente efecto:

Cuando conducen S_1 - S_4 aplican tensión positiva a la carga: $V_{AB} = +V_G$

Cuando conducen S_1 - S_2 aplican tensión nula a la carga: $V_{AB} = 0$

Cuando conducen S_2 - S_3 aplican tensión negativa a la carga: $V_{AB} = -V_G$

Cuando conducen S_3 - S_4 aplican tensión nula a la carga: $V_{AB} = 0$

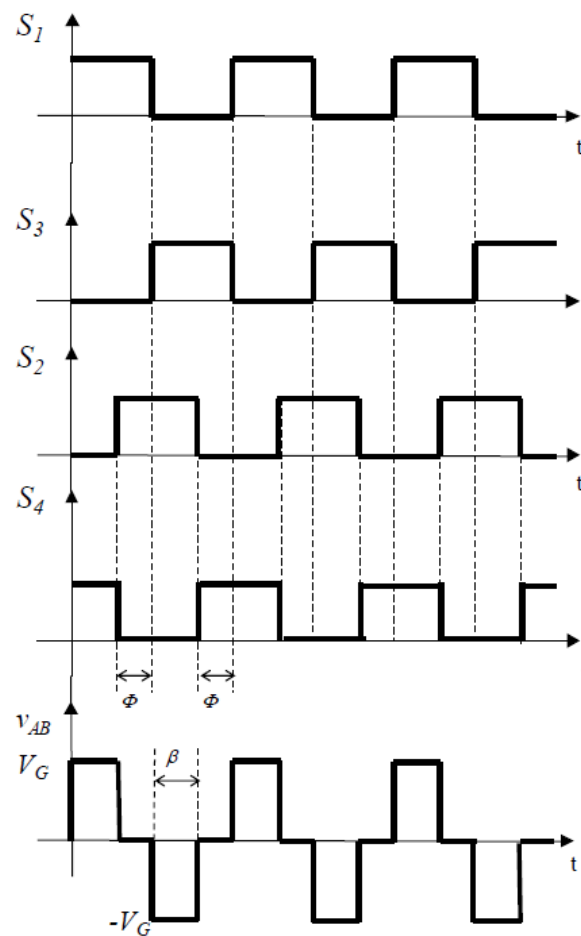


Figura 6 [4]

2.3.2 Modulación PWM Bipolar.

En esta modulación, solo existen dos posibles valores de V_{AB} : $+V_G$ y $-V_G$

Alternando los valores de $+V_G$ y $-V_G$ con intervalos de tiempo diferentes se puede variar el valor medio de cada intervalo T como se muestra en la **Figura 7** original de [4], en la cual se

observa que, para un mismo periodo T , si aumentamos el tiempo que la señal cuadrada permanece a nivel alto, aumenta el valor de tensión media.

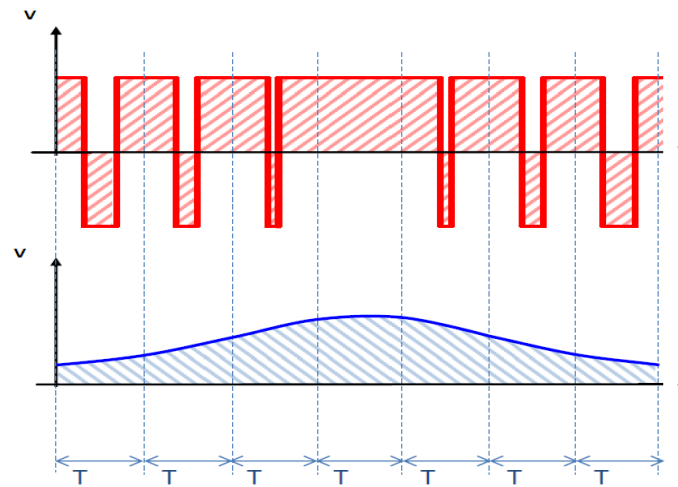


Figura 7 [4]

Para generar una señal cuadrada de ciclo de trabajo variable se comparan dos señales, una de ellas es un seno (señal moduladora) y la otra una señal triangular (señal portadora), y de esta forma surgen dos parámetros importantes que son la modulación en amplitud y la modulación en frecuencia.

El índice de modulación en amplitud es el resultado de dividir la amplitud de la señal moduladora entre la amplitud de la señal portadora. Variando este parámetro conseguimos variar la tensión que llega a nuestra carga. Por otro lado está el índice de modulación en frecuencia, que es el resultado de dividir la frecuencia de la portadora, entre la frecuencia de la moduladora. Aumentar el índice de modulación en frecuencia favorece la aparición de armónicos a frecuencias mayores, en el presente trabajo este parámetro permanecerá constante, pero adoptará un valor alto para evitar la aparición de armónicos a bajas frecuencias.

En la **Figura 8** original de [4] aparece representada en color verde la señal portadora, en color azul la señal moduladora, y en color rojo la señal que resulta de comparar ambas señales. Si la señal moduladora supera a la señal portadora se obtendrá $V_{AB} = +V_G$ si por el contrario la señal moduladora tiene un valor inferior al de la portadora se obtendrá $V_{AB} = -V_G$.

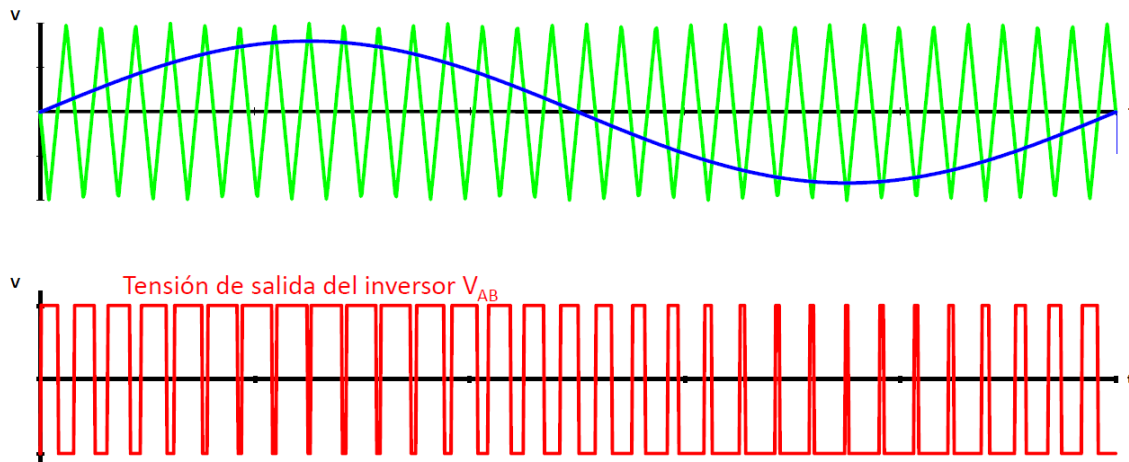


Figura 8 [4]

2.3.3 Contraste entre modulaciones

Existe una gran diferencia entre ambas modulaciones que puede suponer un ahorro, y es que en la modulación por desplazamiento de fase aparecen armónicos muy cercanos a la frecuencia de la componente fundamental (50 Hz), concretamente, el armónico más cercano a la componente fundamental se produce a una frecuencia de 150 Hz, como consecuencia se produce una distorsión armónica total (DAT) elevada y una pequeña distorsión en la tensión de salida resultante como se observa en la **Figura 9** (Obtenida de PSIM) en la cual aparece reflejada la forma de onda de la tensión de salida de convertidor inversor para un valor $C=3.3 \mu\text{F}$ ($f_c=69 \text{ Hz}$) y para un valor $C=10 \mu\text{F}$ ($f_c=121 \text{ Hz}$)

Para ilustrar este efecto, en primer lugar se introdujo un filtro LC a una frecuencia de resonancia de aproximadamente 67 Hz y posteriormente se introdujo un filtro menos restrictivo a una frecuencia de resonancia de 121 Hz. En el primer caso, el filtrado se lleva a cabo con éxito, sin embargo en el segundo caso se observa la distorsión producida por la presencia de armónicos que no han sido correctamente filtrados.

Por otro lado la modulación bipolar presenta armónicos a frecuencias entorno a 1000 Hz, que es el valor de la frecuencia de modulación, de tal manera que el filtrado resulta más sencillo y no es necesario usar un filtro tan restrictivo con una frecuencia de corte tan cercana a la frecuencia de la componente fundamental que se quiere dejar pasar.

De lo descrito en los párrafos anteriores se deduce que la modulación bipolar es más ventajosa ya que:

- No necesita un filtrado tan restrictivo y por tanto hay ahorro económico porque los componentes reactivos son más pequeños.
- La presencia de armónicos se produce a altas frecuencias luego la onda que se genera a la salida es un seno casi perfecto.

-Supone un ahorro de recursos del microcontrolador Tiva C Series TM4C1294-XL, como se verá en el **capítulo 4**.

Cabe destacar que también existe una desventaja para la modulación PWM Bipolar y es que existen pérdidas por conmutación en los transistores, que son mucho más elevadas que en el caso de la modulación por desplazamiento de fase, pero cuyo valor no repercute si tenemos en cuenta las pérdidas globales.

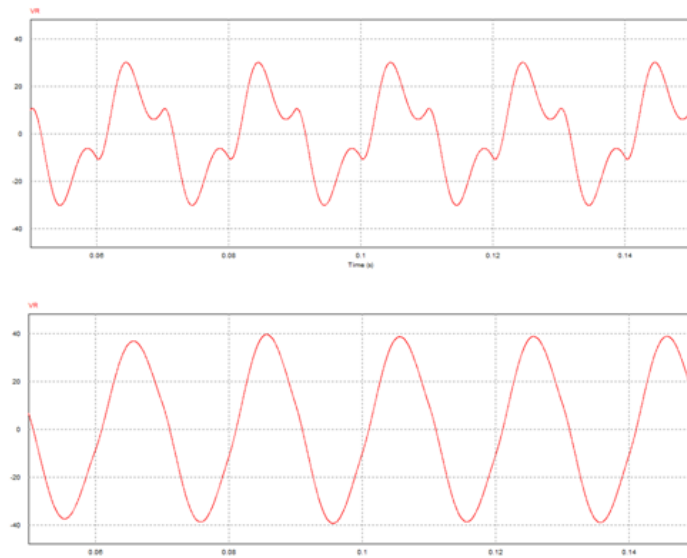


Figura 9

(Imagen de arriba C=3.3 uF, imagen de abajo C=10 uF)

2.4 Análisis estático

En este apartado, se realizará un análisis teórico para comprender el funcionamiento del sistema y poder contrastarlo con los resultados de las mediciones hechas sobre el sistema real.

2.4.1 Análisis estático en modulación por desplazamiento de fase

En primer lugar, se llevará a cabo un análisis teórico en el sistema modulado por desplazamiento de fase, cuya explicación aparece reflejada en el **apartado 2.3.1**, finalmente, a la salida del puente inversor se obtenía una tensión con el siguiente aspecto:

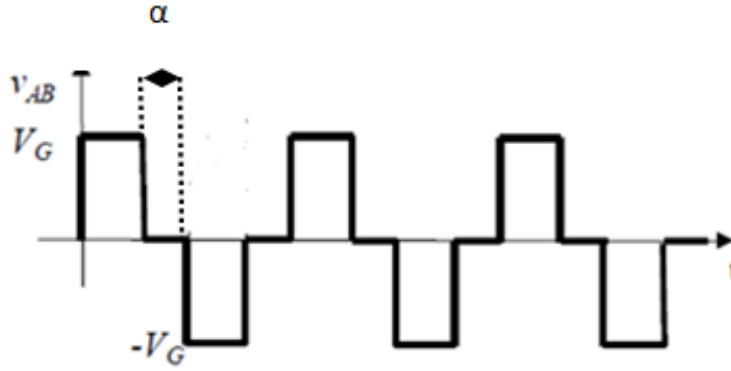


Figura 10 [4]

Donde V_G es la tensión continua de entrada al puente y α es el desplazamiento de fase seleccionado.

Según la teoría de Fourier, una función periódica puede ser descompuesta como la suma de varias funciones senoidales a diferentes frecuencias y amplitudes, por tanto, la función representada en la **Figura 10**, puede ser descompuesta para poder ser tratada. Al descomponer dicha función se obtiene la siguiente serie:

$$v_{AB}(t) = \sum_{n=1,3,5,\dots}^{\infty} \frac{4V_G}{\pi} \cdot \frac{1}{n} \cdot \cos(n\alpha) \sin(n\omega t)$$

Los armónicos que componen dicha función por tanto aparecerán a frecuencias del orden de $n\omega t$ tomando n un valor impar desde 1,3,5 hasta infinito.

A continuación será necesario una etapa de filtrado para conseguir aislar la frecuencia del armónico fundamental, de esta forma aislaremos la señal a 50Hz despreciando armónicos a frecuencias de 150Hz, 250Hz...

Como se expicó en el **apartado 2.3.3**, en el caso de la modulación por desplazamiento de fase, los armónicos se encuentran a frecuencias cercanas a la frecuencia de la componente fundamental (50Hz), luego se necesita un filtro cercano a dicha frecuencia para conseguir una distorsión armónica total baja.

En primer lugar se va a comprobar cuál sería el efecto conseguido empleando un filtro LC con una reactancia de 520 mH y un condensador de 10 uF.

Para calcular la f_c del filtro se usa la siguiente expresión:

$$f_{corte} \approx \frac{1}{2\pi\sqrt{L \cdot C}}$$

Sustituyendo el valor de la reactancia y del condensador se obtiene:

$$f_c = 69.79 \text{ Hz}$$

A partir de esta frecuencia el filtro comenzará a atenuar las señales, siendo la amplitud de la tensión de salida $V_o = V_{AB}(t) * \left| \frac{V_o}{V_{ab}}(j\omega) \right|$ donde el término $\left| \frac{V_o}{V_{ab}}(j\omega) \right|$ corresponde con el módulo de la ganancia de filtro que expresa el grado de atenuación y que tiene el valor dado por la siguiente expresión:

$$\left| \frac{v_o}{v_{AB}}(j\omega) \right| = \frac{1}{\sqrt{(1 - L \cdot C \cdot \omega^2)^2 + \left(\frac{L}{R} \cdot \omega \right)^2}}$$

Como se observa, el grado de atenuación depende de la frecuencia.

Para los cálculos teóricos de este apartado se utilizaran los siguientes valores de los componentes del circuito:

Tabla 1

L	520 mH
C	10 uF
V_G	15 V
α	45°
R	1200 Ω

Obteniéndose como resultado lo siguiente:

Tabla 2

N	Frecuencia (Hz)	$V_{AB}(V)$	Ganancia del filtro	$V_o (V)$
1	50	$\frac{4}{\pi} * V_G * \frac{1}{n} * \cos(n\alpha) = 13.5$	1.98	26.73
3	150	$\frac{4}{\pi} * V_G * \frac{1}{n} * \cos(n\alpha) = 4.5$	0.27	1.215
5	250	$\frac{4}{\pi} * V_G * \frac{1}{n} * \cos(n\alpha) = 2.7$	0.08	0.216

Como se observa en la **Tabla 2**, la componente fundamental (señal a 50Hz) no solo no muestra atenuación sino que además se ve amplificada casi al doble, esto es debido a que la frecuencia de corte del filtro es muy cercana a la frecuencia del armónico fundamental. Por otro lado se ve que el armónico de orden 3 y de orden 5 sufren una gran atenuación haciendo dichos valores casi despreciables frente al valor de amplitud del primer armónico. Para observar si el filtrado es bueno se calculará la distorsión armónica total (DAT) :

$$DAT = \frac{\sqrt{V_{o3}^2 + V_{o5}^2}}{V_{o1}} * 100 = \frac{\sqrt{1.215^2 + 0.216^2}}{26.73} * 100 = 4.61\%$$

La distorsión armónica total es baja luego el filtrado se ha llevado a cabo con éxito, sin embargo, colocando el filtro a una f_c de 67 Hz se produce una amplificación del primer armónico debido a su proximidad con la frecuencia de resonancia del filtro, como no se pretende conseguir este efecto, se va a cambiar de condensador para elevar la frecuencia de corte del filtro y de esta manera se verá claramente la diferencia entre modulaciones descrita en el **apartado 2.3.3**.

Se ha escogido por tanto, un filtro LC con un valor inductivo igual que el anterior y una capacidad de 3.3 μF de tal forma que la frecuencia de corte será:

$$f_c = \frac{1}{2\pi\sqrt{LC}} = 121.5 \text{ Hz}$$

A continuación se calculará el valor de la amplitud de la tensión tras la atenuación del nuevo filtro y la DAT resultante:

Tabla 3

N	Frecuencia (Hz)	$V_{AB}(V)$	Ganancia del filtro	$V_o (V)$
1	50	$\frac{4}{\pi} * V_G * \frac{1}{n} * \cos(n\alpha) = 13.5$	1.18	15.93
3	150	$\frac{4}{\pi} * V_G * \frac{1}{n} * \cos(n\alpha) = 4.5$	1.5	6.75
5	250	$\frac{4}{\pi} * V_G * \frac{1}{n} * \cos(n\alpha) = 2.7$	0.3	0.81

$$DAT = \frac{\sqrt{V_{o3}^2 + V_{o5}^2}}{V_{o1}} * 100 = \frac{\sqrt{6.75^2 + 0.81^2}}{15.93} * 100 = 42.67\%$$

Como resultado se obtiene una distorsión armónica total que aumenta considerablemente tras la implantación del filtro con una f_c superior, luego lo que se verá a la salida del convertidor en este caso no será una tensión senoidal perfecta, sino que se apreciará una distorsión. En los

resultados recopilados en la **Tabla 3** se aprecia que el primer armónico mantiene un valor similar al que tenía antes de sufrir el efecto del filtro, sin embargo, el segundo armónico que debería verse atenuado, se ve amplificado por un factor superior al del primer armónico, esto es debido a que la frecuencia de resonancia del filtro en este caso es más próxima a la frecuencia del segundo armónico. En la **figura 11** se muestra un contraste entre la tensión de salida para el condensador de 3.3 uF (Imagen de la izquierda) y para el de 10 uF (Imagen de la derecha) y claramente se aprecia el efecto inducido.

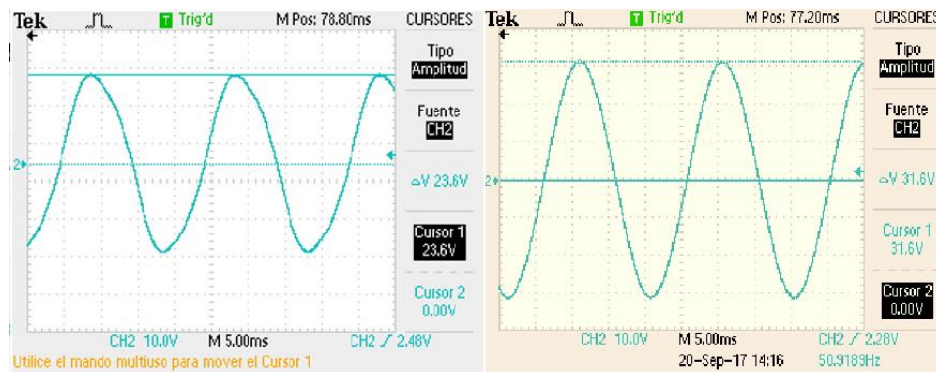


Figura 11

2.4.2 Análisis estático en modulación por PWM bipolar

En segundo lugar se llevará a cabo el análisis estático para la modulación bipolar cuya explicación aparece reflejada en el **apartado 2.3.2**, cabe destacar que en este caso, el análisis estático se llevará a cabo teniendo en cuenta el filtro cuya $f_c = 121.5 \text{ Hz}$. Podría haberse utilizado un filtro con una frecuencia de corte muy superior ya que teóricamente el primer armónico se situará a frecuencias del orden de la frecuencia de modulación (1000 Hz), sin embargo, se ha optado por utilizar este filtro para poder ilustrar de esta forma el efecto que tiene un mismo filtro para ambas modulaciones y en el caso de usar un filtro a f_c muy superior, el efecto sobre el circuito modulado por desplazamiento de fase sería desastroso.

La tensión a la salida del inversor, reflejada en la **figura 8**, puede descomponerse como la suma de varios armónicos, que posteriormente serán filtrados al igual que en la estrategia de modulación por desplazamiento de fase, para obtener el armónico fundamental a 50 Hz.

Para calcular la amplitud de los armónicos es necesario calcular primero el índice de modulación en amplitud y el índice de modulación en frecuencia, antes de comenzar dicho calculo, es necesario explicar un detalle. En el **apartado 2.3.2** se ha hecho referencia a dos

señales que resultan importantes, la señal moduladora y la señal portadora, gracias a la comparación entre ambas señales se consigue generar la forma de onda de la **figura 8** a la salida del puente. En el caso objeto de estudio, esta forma de modular ha sido sustituida ya que al ser controlada mediante microcontrolador, ya tenemos una salida especial que realiza dicha función y solo habría que jugar variando el ciclo de trabajo de dicha salida para generar la PWM bipolar.

Teniendo en cuenta lo descrito anteriormente, se calcularán los parámetros más característicos del sistema:

En primer lugar se fija una $m_a = 1$, es el resultado de dividir la amplitud de la señal moduladora entre la amplitud de la señal portadora como se muestra en la siguiente expresión:

$$m_a = \frac{A_{sin}}{A_{tri}}$$

Por otro lado hay que calcular la modulación en frecuencia cuyo valor es el que se expresa en la siguiente ecuación:

$$m_f = \frac{f_{tri}}{f_{sin}} = \frac{1000 \text{ Hz}}{50 \text{ Hz}} = 20$$

Se ha escogido como frecuencia de la señal triangular 1000 Hz debido a que en el programa, el timer cuya interrupción hace variar el ciclo de trabajo va a esa frecuencia. Por otro lado, la frecuencia de la señal moduladora es 50 Hz debido a que es la frecuencia que se pretende obtener para la señal de salida del convertidor inversor.

A continuación se deben buscar los coeficientes que van a determinar la amplitud de los tres primeros armónicos, para ello hay que situarse en la tabla de amplitudes normalizadas para modulación PWM Sinusoidal Bipolar que aparece ilustrada en la **Figura 12**, y tenemos en cuenta una modulación en amplitud de 1.

Amplitudes normalizadas V_n/V_{max} para modulación PWM Sinusoidal Bipolar										
ma	0,10	0,20	0,30	0,40	0,50	0,60	0,70	0,80	0,90	1,00
n=1	0,10	0,20	0,30	0,40	0,50	0,60	0,70	0,80	0,90	1,00
n= mf	1,27	1,24	1,20	1,15	1,08	1,01	0,92	0,82	0,71	0,60
n= mf+-2	0,00	0,02	0,03	0,06	0,09	0,13	0,17	0,22	0,27	0,32

Figura 12

El valor de amplitud de cada armónico será igual al valor de la tensión continua que aplicamos a la entrada del puente inversor, multiplicado por el correspondiente valor de amplitud normalizada, de manera que se obtienen los siguientes valores:

Tabla 4

N	Frecuencia (Hz)	$V_{AB}(V)$	Ganancia del filtro	$V_o (V)$
1	50	$1*15=15$	1.18	17.7
3	$(mf-2)*50=900$	$0.6*15=9$	0.019	0.171
5	$mf*50=1000$	$0.32*15=4.8$	0.015	0.072

Obteniéndose la siguiente distorsión armónica total:

$$DAT = \frac{\sqrt{V_{o3}^2 + V_{o5}^2}}{V_{o1}} * 100 = \frac{\sqrt{0.019^2 + 0.015^2}}{17.7} * 100 = 0.136 \%$$

Si se compara la DAT obtenida para el filtro con condensador de 3,3 uF en la modulación por desplazamiento de fase y en la modulación bipolar, se aprecia de una forma muy clara que la modulación bipolar es mucho más eficaz con una DAT de 0.136% frente a un 42,67% obtenido en desplazamiento de fase, obteniéndose una tensión de salida del convertidor inversor como la que se muestra en la **figura 13**.

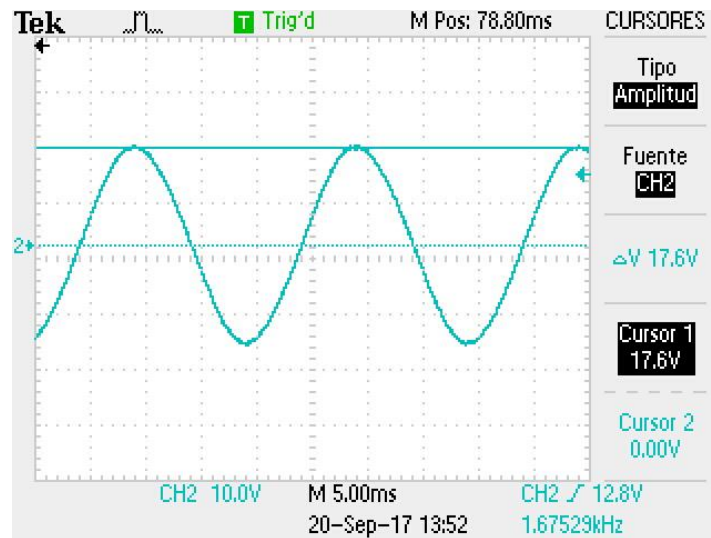


Figura 13

2.5 Control del convertidor inversor puente completo

En este apartado se presenta la estrategia de control tanto en lazo abierto como en lazo cerrado.

2.5.1 Control en lazo Abierto

El control en lazo abierto de un sistema no asegura que la variable a controlar mantenga el valor deseado ante perturbaciones producidas en las variables de las que depende el circuito, de tal forma que si se produce una variación en una de ellas, la señal de control (el desfase en el caso de la modulación por desplazamiento de fase o la modulación en amplitud en el caso de la modulación PWM Bipolar) no se modificará para contrarrestar dicha perturbación.

Este tipo de sistemas se caracteriza por ser sistemas sencillos, en los cuales no es de vital importancia la presencia de un sistema de control que asegure la inmunidad ante algunas perturbaciones porque no produzcan un efecto significativo en la magnitud de salida del sistema.

Un ejemplo de control en lazo abierto es el llenado de un depósito, si la llave de paso se abre el depósito comenzará a llenarse, sin embargo cuando este llegue a su capacidad máxima, la llave seguirá abierta a no ser que el usuario la cierre.

En nuestro sistema, la llave de paso será el desfase o la modulación en amplitud, que son las variables que se pueden modificar a través de la interfaz con el usuario (variables de control), esta modificación dará lugar a una tensión de amplitud y valor eficaz variables a la salida del convertidor. Al no existir realimentación la salida depende de forma directa de la entrada como aparece reflejado en la **figura 14**.

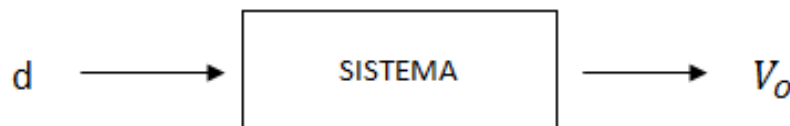


Figura 14

2.5.2 Control en lazo Cerrado

Un sistema de control en lazo cerrado es aquel en el cual, las variables de las cuales depende el sistema se regulan de forma automática dentro de cierto rango para asegurar la estabilidad del sistema, es decir, la señal de control sigue en lo posible a la de referencia consiguiendo que el efecto de las perturbaciones externas sea el menor posible, también se denominan sistemas retroalimentados.

Un ejemplo de control en lazo cerrado sería por ejemplo el control de nivel de un depósito, este control se efectúa sobre la válvula que controla el llenado, de tal forma que si el depósito llega a su capacidad máxima, de forma automática la válvula se cerrará. Siempre se tendrá noción del valor de los sensores que controlan el nivel del agua y en función de estos valores la válvula estará más o menos abierta.

En el control por lazo cerrado es necesario hacer un sensado de la variable que queremos controlar. En nuestro caso, la variable a controlar es la amplitud de la tensión de salida y nuestro sensor será el ADC integrado en el microcontrolador Tiva C Series EK-TM4C1294XL.

Se ha optado por hacer un regulador integrador básico para eliminar el error en régimen permanente, cuyo esquema se muestra en la **figura 15**.

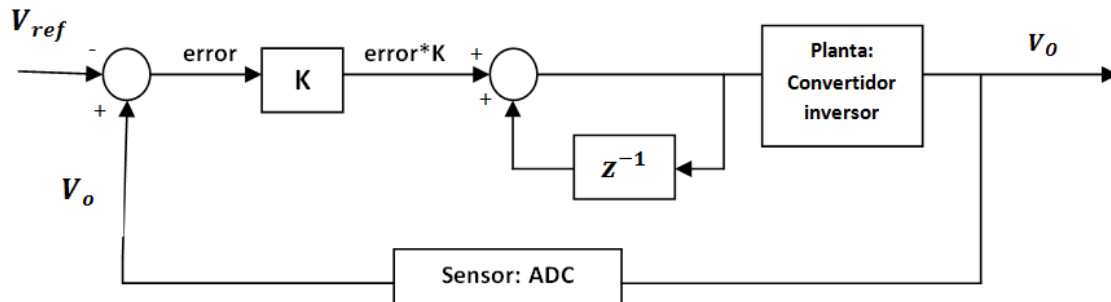


Figura 15

El valor de K se ha obtenido de forma experimental para garantizar que el sistema es estable en todo el rango de funcionamiento y que la amplitud de la tensión de salida se ajusta al valor de referencia en un tiempo menor a 0.5 s. El regulador utilizado es un regulador sencillo, la respuesta del sistema no será muy rápida pero sí estable.

PROTOTIPO CONVERTIDOR INVERSOR PUENTE COMPLETO

En este capítulo se procederá con el análisis del prototipo real del convertidor inversor de puente completo y todo el hardware que ha sido necesario para su puesta en marcha.

3.1 Puente de transistores

El puente inversor fue proporcionado por el grupo de Sistemas Electrónicos de Potencia de la Universidad Carlos III de Madrid para poder utilizarlo en el presente trabajo. En la **figura 16** se muestra una imagen de dicho componente, indicando cual es la función de cada una de las entradas.

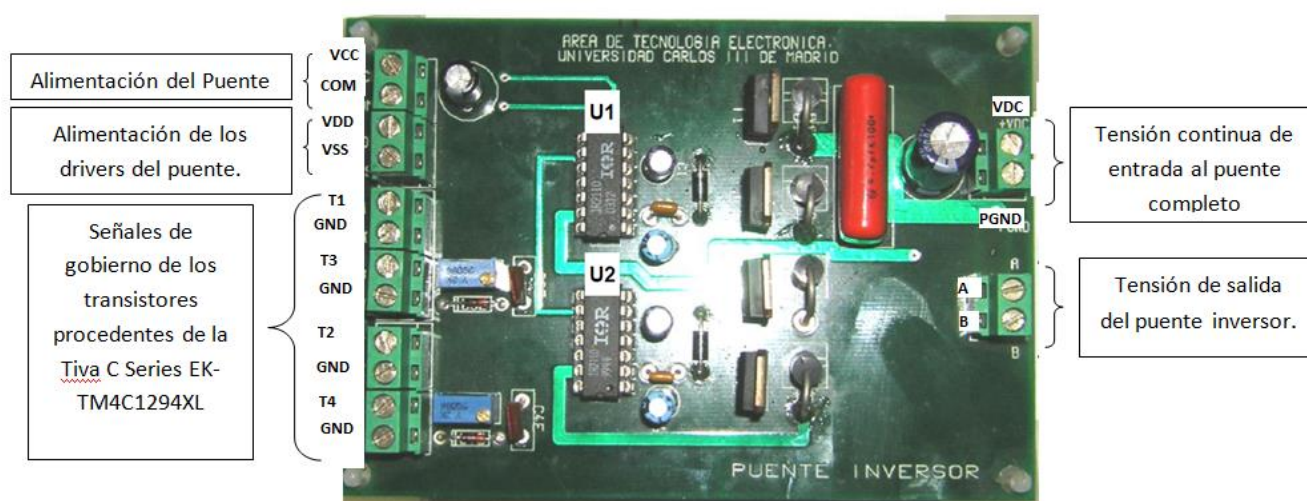


Figura 16

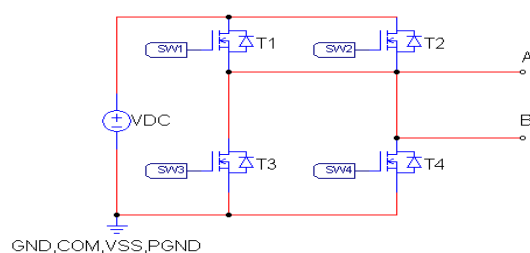


Figura 17

Todos los pines del puente inversor cuyo conexionado describiremos a continuación aparecen reflejados en la **Figura 16**.

El puente inversor se alimenta con 15 V a través de VCC, sus drivers se alimentan a 5 V a través de VDD, VDC corresponde con la tensión de entrada del puente inversor, y en este caso va a ser alimentada con 15 V. Por otro lado, cada uno de los terminales nombrados anteriormente posee su tierra, COM, VSS y PGND irán conectados todos a la tierra procedente del micro controlador Tiva.

Los terminales T1,T3,T2 y T4 corresponden con las señales de gobierno de los transistores MOSFET que van integrados en la placa (puente inversor), la señal de gobierno de dichos transistores procede del microcontrolador. Cada uno de ellos tiene su tierra, para evitar la aglomeración de cables se decidió puentearlos entre sí y llevar un solo cable a tierra.

Por último, los terminales A y B corresponden con la tensión de salida del puente inversor, una tensión diferencial que no va referida a tierra y que posteriormente será filtrada mediante un filtro LC.

3.2 Tiva C Series EK-TM4C1294XL Texas Instruments

Tiva C Series EK-TM4C1294XL es un microcontrolador fabricado por Texas Instruments mediante el cual se llevará a cabo el control del inversor de puente completo. Será el encargado de generar los pulsos de gobierno de los transistores del puente inversor, de realizar el muestreo de la tensión de salida (en forma de seno) para hallar un valor máximo en el caso de la modulación por desplazamiento de fase y un valor eficaz en el caso de la modulación PWM Bipolar y poder conocer así la amplitud de la onda, además, el micro se encargará de llevar a cabo el control para alcanzar una tensión de salida previamente asignada de forma automática.

La tarjeta Tiva C Series EK-TM4C1294XL (**Figura 18**) presenta numerosas funciones, a continuación se hará un breve resumen de aquellas que se emplearán en el presente trabajo:

- 8 Salidas PWM: Este micro dispone de 8 salidas PWM de las cuales se emplearán 4 para la modulación por desplazamiento de fase, y 2 para la modulación por PWM bipolar.
- Comunicación Serie: Dispone de 8 UARTs (Universal Asynchronous Receiver/Transmitter) de las cuales solo se usará 1 para mostrar el menú a través del cual el usuario podrá interactuar con el micro y con ello modificar ciertos valores en el inversor.
- Puerto USB: A través del puerto USB se alimenta el micro controlador además de llevar a cabo la programación en lenguaje C y poder depurar el funcionamiento del sistema.
- Interfaz con el Usuario: Dispone de 4 LEDs y 2 Switch para interactuar con el usuario, aunque en el presente trabajo no será necesario su uso.

- 2x12 bits ADC converter: Dispone de dos convertidores de analógico a digital, en este trabajo solo se utilizará uno para llevar a cabo la lectura de la tensión de salida del inversor y poder de esa forma controlar el sistema.
- Frecuencia del Reloj: La frecuencia del reloj en el caso de la modulación bipolar es de 120 MHz, en la modulación por desplazamiento de fase será de 2,86 MHz debido a la necesidad de reducir la frecuencia de reloj para producir señales PWM a 50Hz.

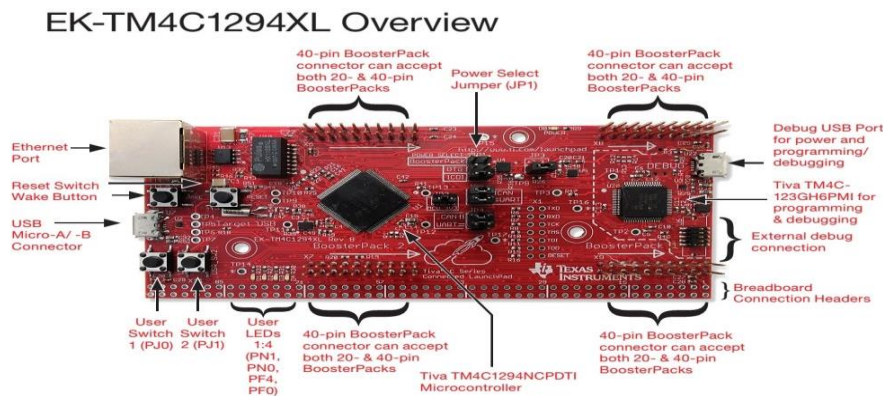


Figura 18

3.3 Adaptación de la tensión de salida

Para llevar a cabo la lectura de la tensión de salida mediante el ADC, cuya estrategia será explicada en el **capítulo 4**, es necesario llevar a cabo previamente un tratamiento de la señal.

La tensión de salida del inversor alcanza valores de pico de hasta 24 V y el ADC del micro controlador soporta valores de tensión máxima de 3.3 V, además, al ser una señal alterna la tensión adoptará valores positivos en un semiperiodo y negativos el tiempo restante, y la tarjeta EK-TM4C1294XL no soporta tensiones negativas, por tanto, antes de introducir la tensión en el pin de lectura del ADC fue necesario un acondicionamiento de la señal hasta conseguir una tensión apta para ser introducida al ADC.

En los **apartados 3.3.1 y 3.3.2** se explicarán las dos etapas de acondicionamiento de la señal de una forma más detallada e ilustrativa.

3.3.1 Divisor resistivo

En primer lugar, para no superar la tensión máxima admitida por el ADC (3.3 V) se ha aplicado un divisor resistivo en cada extremo de la R_{carga} para disminuir la amplitud de la tensión de salida hasta alcanzar un valor de pico no superior a 1 V. Se llamará V_{ad620} a la diferencia de tensión entre la pata inversora y la no inversora, que es la tensión que se va a aplicar en la

siguiente etapa en la cual interviene el circuito integrado AD620. En la **Figura 19** aparece representado el esquema eléctrico de la etapa de acondicionamiento previamente descrita.

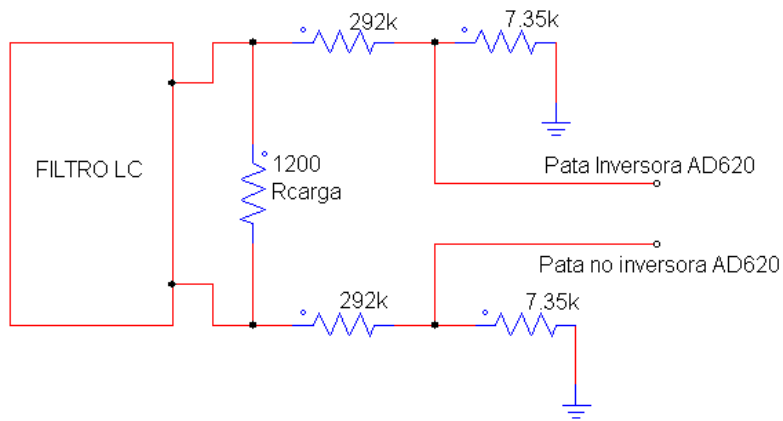


Figura 19

En la siguiente imagen se observan dos formas de onda, la amarilla se corresponde con la tensión de salida del convertidor inversor con el condensador de 10uF, por otro lado la azul se corresponde con la tensión Vad620, es decir, la tensión de salida una vez aplicado el divisor resistivo y antes de pasar por el integrado AD620.

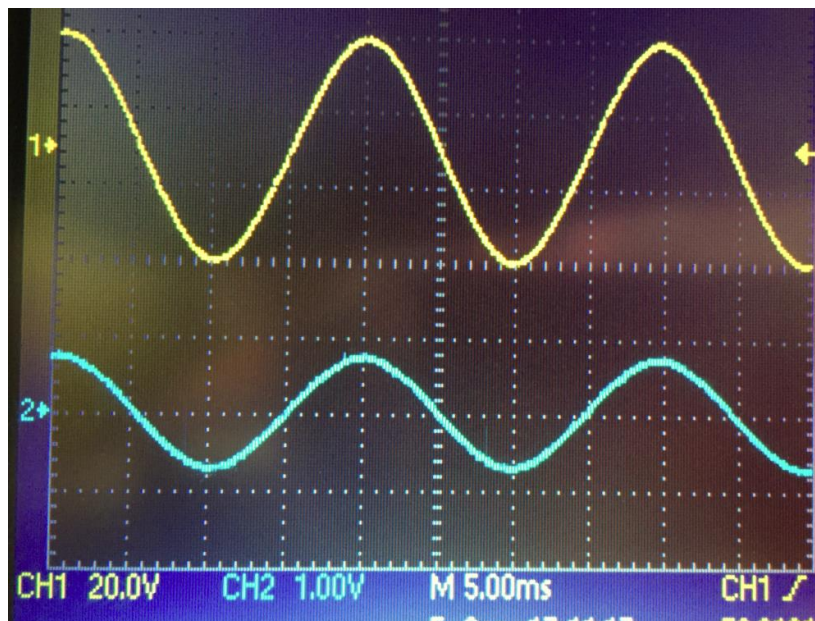


Figura 20

En esta imagen se puede apreciar el efecto del divisor resistivo, que hace que la forma de onda amarilla que alcanza aproximadamente 60 V pico-pico, se transforme en la forma de onda azul que no llega a alcanzar los 2 V pico-pico, tensión que sería apta para el micro controlador si no tuviese tensiones negativas, por ello es necesario incluir la etapa de adaptación que se explica en el siguiente apartado.

3.3.2 Eliminar tensiones negativas

En esta etapa se va a tratar la señal procedente del divisor resistivo para conseguir eliminar las componentes negativas. Para alcanzar dicho objetivo se ha optado por introducir una componente continua a la señal procedente del divisor resistivo, es decir, introducir un offset.

Tras probar numerosas configuraciones se optó por utilizar el circuito integrado AD620 [5] ya que este integrado permite introducir una referencia en una de sus patas, y de esta forma se consigue introducir un offset. La alimentación del integrado se lleva a cabo de forma simétrica con $\pm 10\text{ V}$. El esquema del circuito integrado AD620 se presenta en la **figura 21**.

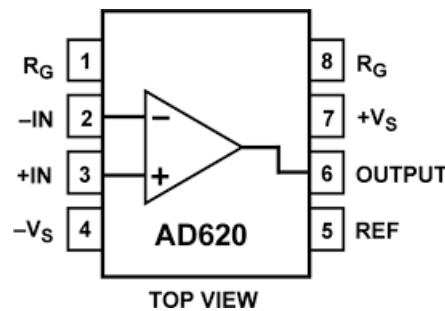


Figura 21

A continuación se describirá la función de cada una de sus entradas/salidas:

- 1 y 8: Estos dos terminales denominados R_G son los terminales de ajuste de ganancia, la ganancia del circuito es la siguiente:

$$G = 1 + \frac{50\text{ k}\Omega}{R_G}$$

Puesto que en nuestro caso la amplitud ya ha sido controlada mediante el divisor resistivo, sería conveniente que la ganancia en este caso fuese de 1 para evitar más modificaciones en la tensión de salida. Para lograr que la ganancia sea '1' se ha conectado entre los terminales R_G una resistencia de $10\text{ M}\Omega$.

- 2: Entrada inversora del amplificador operacional.
- 3: Entrada no inversora del amplificador operacional.
- 4: Alimentación negativa (-10 V)

- 5: Este es el terminal que servirá para ajustar el offset, el esquema del circuito empleado es el siguiente:

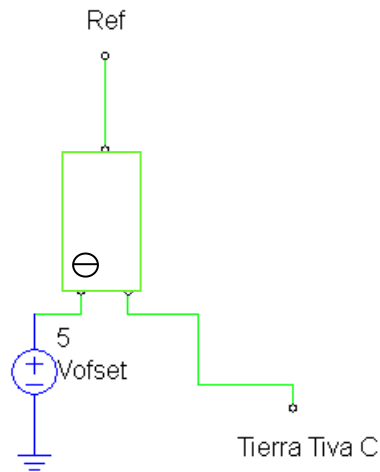


Figura 22

El potenciómetro fue introducido para ajustar un offset, aunque una vez ajustado permanecerá fijo.

- 6: Salida del amplificador operacional (tensión adaptada)
- 7: Alimentación positiva (+10 V)

En la **figura 23** se observa la forma de onda obtenida a la salida del circuito integrado AD620, gracias a la acción de dicho circuito hemos conseguido eliminar las tensiones negativas. Si se observa el valor del cursor 1, la tensión no supera los 2.5 V, ya que introducir tensiones superiores a 3.3 V sería muy perjudicial para el micro controlador, además se aprecia que la tensión pico-pico es de 1.4 V aproximadamente según la escala CH2 1.00 V/div.

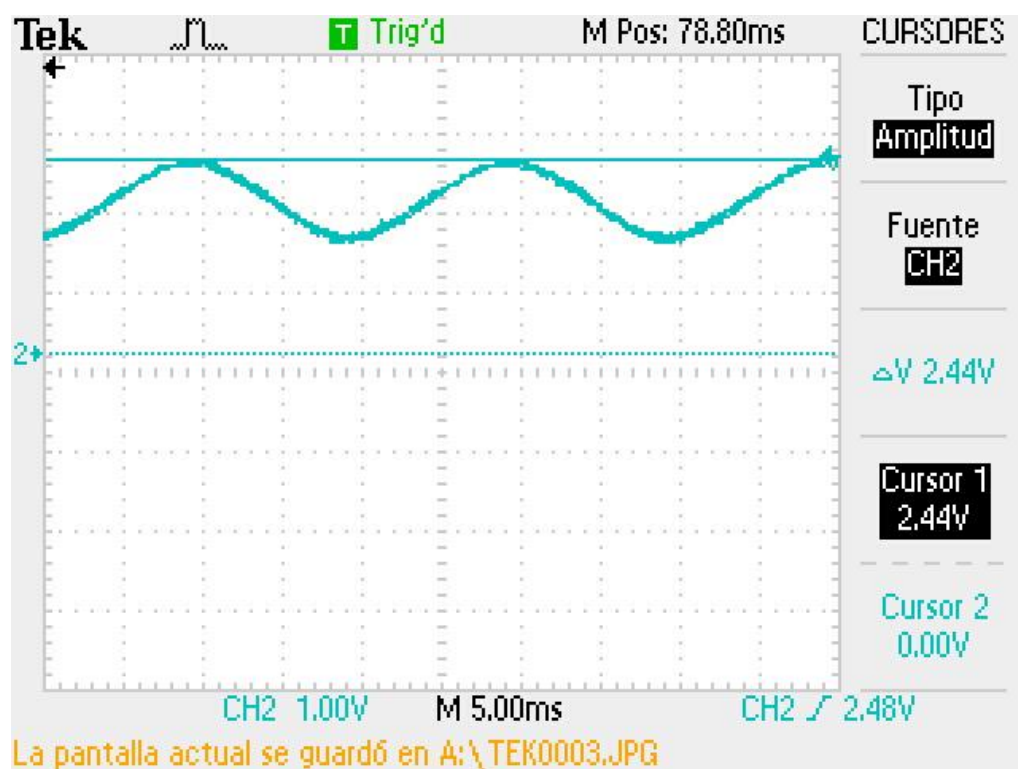


Figura 23

3.4 Componentes para el montaje del circuito.

Para montar el convertidor inversor se han empleado diversos componentes eléctricos, la mayoría de ellos proporcionados por el departamento de Electrónica de Potencia de la Universidad Carlos III. A continuación se presentarán todos ellos:

Para llevar a cabo el filtrado de la tensión de salida del puente se han empleado un condensador de 3.3 μF y una reactancia de 520 mH. La resistencia de salida del puente inversor tendrá un valor de 1200 Ω .

Para implementar el divisor resistivo se han utilizado cuatro resistencias, dos de ellas de 290 k Ω y otras dos de aproximadamente 7 k Ω .

Para ajustar la ganancia del circuito AD620 se ha utilizado una resistencia de 10 M Ω y para ajustar el ofset un potenciómetro de 10 k Ω . Además se han colocado dos condensadores de 4.7 nF en la alimentación positiva y negativa del AD620 para evitar el ruido.

3.5 Alimentación.

Para la alimentación de todo el sistema fueron necesarias cinco fuentes de tensión.

Las fuentes fueron proporcionadas por el departamento de Electrónica de Potencia de la Universidad Carlos III, y poseen 3 canales cada una, dos de ellos regulables y uno fijo a +5V.

Debido a lo descrito en la línea anterior solo fue necesario el uso de dos fuentes, a continuación se describirá el propósito de cada una de ellas.

En primer lugar fue necesario alimentar el puente inversor a +15 V y sus drivers a +5V, para alimentar el puente a 15 V se utilizó uno de los dos canales regulables de una de las fuentes, el otro se utilizó para introducir la tensión de entrada del puente inversor, que aunque en el presente trabajo será de 15 V también, se decidió independizarla para realizar diversas comprobaciones experimentales. Para alimentar los drivers a +5V se utilizó el canal fijo de +5V. Para todo lo descrito anteriormente se utilizó una de las dos fuentes.

Por otro lado, para el ajuste de offset se utilizó el canal fijo a +5 V de la otra fuente, ajustando el valor de tensión de offset requerido mediante el potenciómetro.

Por último, para alimentar el integrado AD620 a ± 10 V se utilizó esta misma fuente configurada de forma simétrica para obtener dicha tensión, por lo que fue necesario el uso de los dos canales regulables.

De esta forma se utilizaron todos los canales de ambas fuentes, el conexionado aparece reflejado en la **figura 24**.

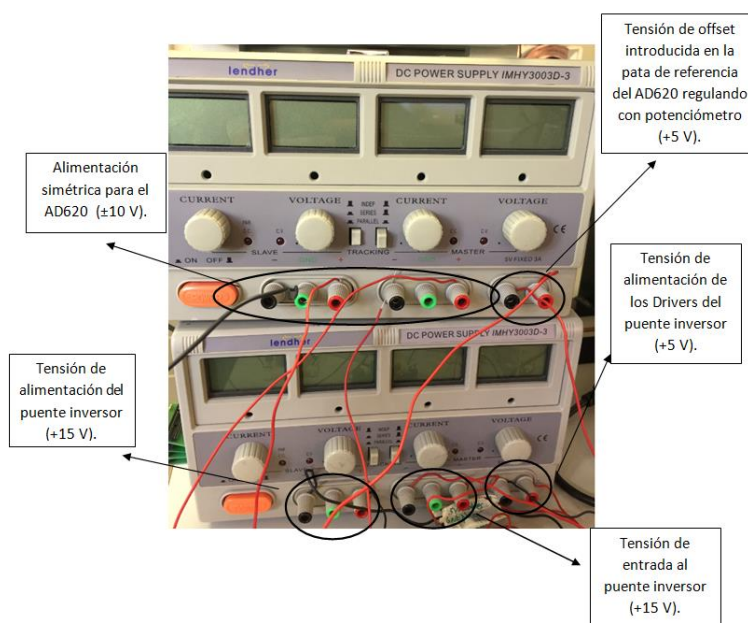


Figura 24

3.6 Conexionado en modulación por desplazamiento de fase

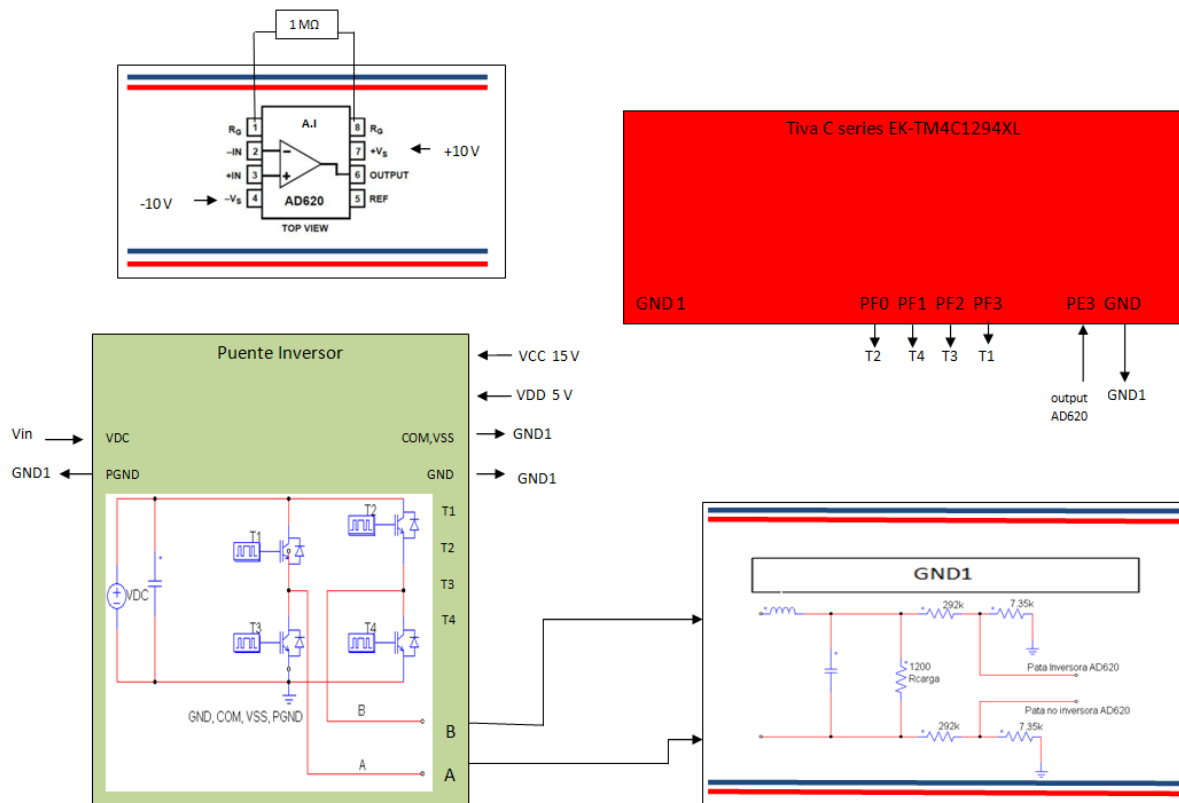


Figura 25

A continuación se lleva a cabo una breve descripción del conexionado del sistema, para que la explicación pueda seguirse más fácilmente, en la **figura 25** se presenta un esquema.

Desde la tarjeta Tiva C Series TM4C1294XL se obtendrán los pulsos de gobierno de los transistores integrados en el puente inversor, luego los pines correspondientes del micro (PFO,PF1,PF2 y PF3) deberán ser conectados a las entradas de los transistores del puente inversor correspondientes (T1, T2, T3 y T4) . Del micro también se obtendrá la referencia (GND 1) que será la referencia de todo el sistema, exceptuando el tramo final del inversor (Tensión de salida) que es una tensión diferencial y no va referida a tierra, La tarjeta Tiva además posee una entrada a través del pin PE3, donde recibirá la señal de salida del AD620 para hacer la lectura a través del ADC de 12 bits.

Por otro lado está el puente inversor, cuya alimentación explicamos en el **apartado 3.1**, todas sus tierras van referidas a GND 1. Su salida (A y B) va conectada a la etapa de filtrado, representada en la Protoboard que en la **Figura 25** aparece en la esquina inferior derecha.

Finalmente, una vez filtrada y atenuada mediante el divisor resistivo, la señal será introducida en la pata inversora y no inversora del AD620, donde se introducirá un offset para evitar las

tensiones negativas. La señal procedente de la pata OUTPUT del AD620 será introducida en el ADC de la tarjeta EK-TM4C1294XL, y de forma digital, se le restará el offset introducido y se multiplicará el valor obtenido por el valor de ganancia aplicada en el divisor resistivo para obtener el valor real de tensión a la salida del inversor. En la **figura 26** se presenta una imagen del circuito real una vez montado.

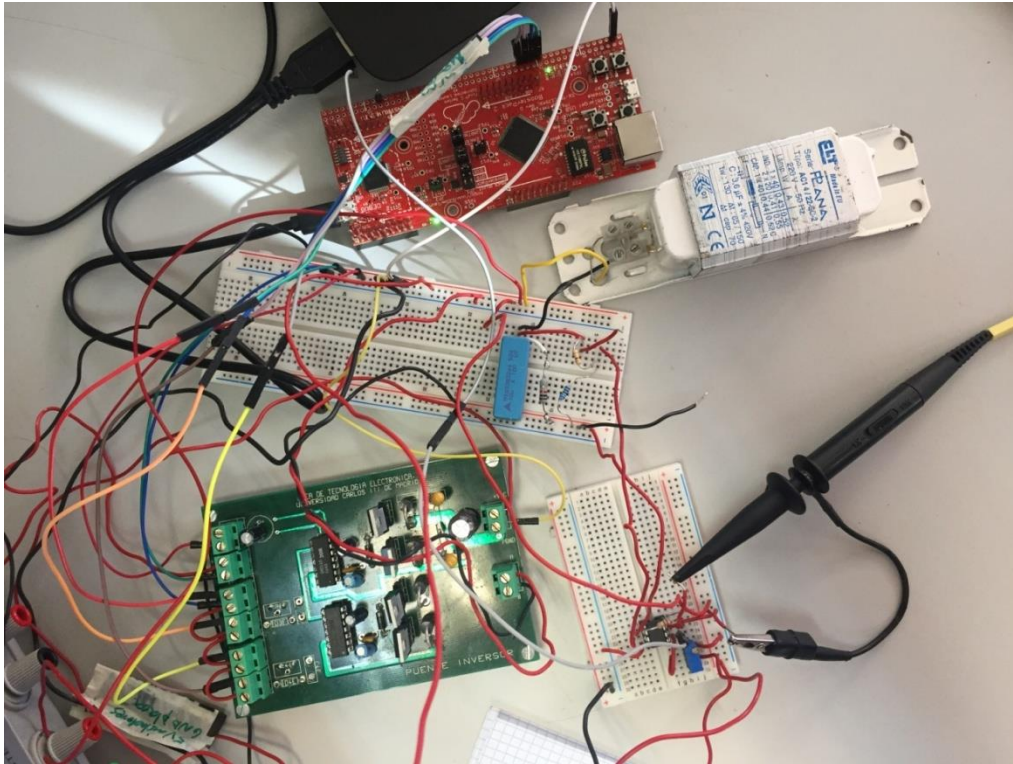


Figura 26

Para la modulación PWM bipolar, el montaje será similar, la diferencia fundamental reside en que en este caso se utiliza un solo generador PWM de la tarjeta Tiva, por tanto únicamente será necesario el uso de dos pines, PF0 y PF1. Esto es debido a que en este caso no es necesario introducir un desfase, la tensión en el pin PFO controlará los transistores T1 y T4 y por otro lado el pin PF1 que es el negado de PF0 controlará los transistores T2 y T3. La estrategia de generación de la señal PWM Bipolar se explicará más detalladamente en el **apartado 4.3.3.2.**

CAPITULO 4

PROGRAMACIÓN DEL SISTEMA.

En el **capítulo 4** se tratarán todas aquellas partes del proyecto que tengan que ver con el desarrollo de software. Este software será programado en código C e implementado en la placa Tiva C Series EK-TM4C1294XL cuya función principal es la de generar los pulsos de control del inversor según la estrategia de control deseada y realizar el control en lazo cerrado en su caso. Cabe destacar, que se decidió realizar un programa distinto para la modulación por desplazamiento de fase y para la modulación PWM Bipolar, debido principalmente a la diferencia en cuanto a la generación de las señales PWM y a su repercusión a nivel hardware.

La placa Tiva C Series se conectará a través de un puerto USB del ordenador y se programará en el entorno de desarrollo integrado del programa Code Composer Studio, que sirve para programar microcontroladores y sistemas empujados de Texas Instruments. Además será necesario el uso de Tera Term, un programa cuya finalidad es establecer la conexión serie entre el ordenador y el micro controlador.

En primer lugar se realizará un resumen de los recursos del microcontrolador que se han utilizado, posteriormente se describirá la finalidad de cada una de las variables utilizadas, la programación necesaria para activar cada periférico y la estrategia para generar ambas modulaciones empleadas. Además se hará una descripción de las funciones utilizadas acompañada de los diagramas de flujo correspondientes.

4.1 Funcionamiento y decisiones de diseño.

La finalidad del presente trabajo es ser utilizado en las prácticas de aquellas asignaturas que estén relacionadas con la Electrónica de Potencia, por tanto, la interfaz gráfica que he desarrollado intenta ser lo más intuitiva posible para que resulte de fácil comprensión.

En dicha interfaz, el usuario podrá interactuar con el micro controlador Tiva C Series EK-TM4C1294XL actuando así sobre el convertidor inversor modificando diversas variables del sistema y viendo cual es la repercusión.

En primer lugar, aparecerá un menú a través de la interfaz del programa Tera Term en el cual se da la opción de control en lazo abierto, control en lazo cerrado y ver características del sistema en ese instante (desfase, modulación en amplitud, tensión de salida etc.). Si el usuario selecciona el modo de control en lazo abierto le aparecerá el menú de este modo de control, en el cual podrá escoger entre variar el desfase de las PWM en el caso de la modulación por desplazamiento de fase, o la modulación en amplitud en el caso de PWM bipolar.

Si por el contrario se selecciona el modo de control en lazo cerrado, el usuario podrá escoger entre varios valores de tensión de salida y el sistema modificará las señales de control de forma automática para alcanzar dicha tensión.

Además, si se desea conocer el valor de la amplitud, del desfase, o de la modulación en amplitud, existe una opción que nos muestra dicha información.

4.2 Recursos empleados

A continuación se presenta una tabla con los recursos del microcontrolador Tiva C series EK_TM4C1294XL empleados:

Tabla 5

PIN	TIPO	FUNCIÓN
PF0	Salida PWM	Control de transistor 2 en el modo desplazamiento de fase, y de T1 y T4 en el modo PWM Bipolar
PF1	Salida PWM	Control del transistor 4 en el modo desplazamiento de fase, y de T2 y T3 en el modo PWM Bipolar
PF2	Salida PWM	Control del transistor 3 en el modo desplazamiento de fase
PF3	Salida PWM	Control del transistor 1 en el modo desplazamiento de fase
-	TIMER A, periódico	Controla la conversión del ADC, el lazo cerrado y más cosas que explicaremos posteriormente.
PA0 (RX0), PA1 (RX1)	PUERTO SERIE UART0	Permite la comunicación serie en ambos sentidos
PE3	Entrada analógica del ADC	Muestrear la tensión de salida para obtener la amplitud

4.2.1 Señales PWM

En el presente trabajo, existen dos estrategias diferentes de modulación, por un lado se encuentra la modulación por desplazamiento de fase explicada en el **apartado 2.3.1** y por otro lado la modulación por PWM bipolar explicada en el **apartado 2.3.2**, dependiendo de la modulación empleada será necesario utilizar más o menos señales PWM.

En el caso de la modulación por desplazamiento de fase es necesario el uso de 4 señales PWM, en el microcontrolador eso se traduce a utilizar dos generadores de PWM, cada generador impondrá en un pin la señal PWM con ciclo de trabajo 50% y en otro pin su complementaria,

además de añadir un desfase entre las señales como se explicará en el **apartado 4.3.3.1**, de esta forma se controlará el disparo de los 4 transistores.

En el otro caso (PWM bipolar) solo serán necesarias 2 señales PWM (1 Generador de señales PWM) ya que cada señal controlará dos transistores. En este caso el ciclo de trabajo de las señales no será del 50% sino que se hará variar entre 0 y 1 a una frecuencia de 1000 Hz tomando valor de una función seno cuya amplitud tomará el valor de la modulación en amplitud deseada, la estrategia de generación de las PWM en el modo bipolar se explicará con más detalle en el **apartado 4.3.3.2**.

4.2.2 Timer

El Timer periódico es una de las partes más importantes del sistema ya que en la interrupción generada cada vez que desborda, se asignarán los diferentes valores del ciclo de trabajo en lazo abierto o cerrado en el modo por desplazamiento de fase, y los diferentes valores de modulación en amplitud en el modo pwm bipolar. Además en la interrupción del Timer también se llevará a cabo la lectura de la tensión de salida y un pequeño algoritmo para mostrar números decimales a través de la UART. La frecuencia del Timer es de 10000 Hz en el caso de la modulación por desplazamiento de fase ya que es importante realizar un muestreo a una frecuencia alta para poder de esta manera obtener un control más fiable sobre la magnitud a medir e implementar mecanismos de seguridad futuros en caso de percibir cualquier valor anormal y 1000 Hz en el caso de la modulación PWM bipolar.

4.2.3 ADC

El convertidor analógico digital de 12 bits será controlado mediante el Timer, cada vez que se produzca una interrupción (frecuencia igual a 10000 Hz o 1000 Hz) se realizará una lectura de la tensión de salida, de esta forma se realizará un muestreo continuo de la tensión de salida almacenando el valor máximo leído que será el valor de la amplitud, aunque en el modo PWM bipolar se hará de forma diferente como explicaremos posteriormente.

4.2.4 UART (Comunicación puerto serie)

El protocolo serie UART es la forma que tenemos de transmitir y recibir datos desde el usuario, hasta el micro controlador y de esta forma poder generar una interfaz e interactuar de forma directa con el convertidor inversor.

La velocidad de transmisión de datos seleccionada es de 115200 baudios/segundo.

Además, para poder establecer la conexión serie entre el ordenador y la placa, se ha requerido el uso de otro programa denominado Tera Term.

4.3 Código

En este apartado se comentarán las diferentes partes del código, la estrategia de generación de las señales PWM en cada tipo de modulación y la configuración de cada uno de sus parámetros, los parámetros de programación del Timer y del ADC y las variables que intervienen en el programa.

4.3.1 Variables globales en modulación por desplazamiento de fase

A continuación se presenta una tabla en la que se describen los nombres, tipos y finalidades de las variables empleadas en la modulación por desplazamiento de fase:

Tabla 6

Nombre	Tipo	Función
g_ui32SysClock	uint32_t	Reloj del Sistema.
Vo	uint32_t	Almacena la conversión del ADC.
Int32IntegerPartT	Int_fast32_t	Parte entera de la amplitud de la tensión de salida.
Int32FractionPartT	Int_fast32_t	Parte decimal de la amplitud de la tensión de salida.
Int32IntegerPartD	Int_fast32_t	Parte entera del ciclo de trabajo.
Int32FractionPartD	Int_fast32_t	Parte decimal del ciclo de trabajo.
fVoant	float	Almacena el valor máximo de tensión en cada periodo (Amplitud)
fVo	float	Valor de conversión del ADC almacenado en variable de tipo float
fVomed[100]	float	Almacena 100 máximos de tensión (Amplitud) para hacer la media aritmética.
sumatensiones	float	Almacena la suma de las 100 amplitudes almacenadas en fVomed[100]
tension	float	Almacena el valor de sumatensiones, para poder poner dicha variable a 0 para la siguiente cuenta.
tensionfinal	float	Valor final de tensión que se mostrará a través de la UART
tensionref	float	En lazo cerrado, tensión que

		se desea alcanzar.
d	float	Desfase a introducir.
Seleccion2 Selec Selecdeadb selecdef	char	Variables de selección, guardan una lectura del teclado para acceder a los diferentes menús.
errorahora	float	Error en cada lectura
errorganancia	float	errorahora multiplicado por la ganancia K
K	float	Ganancia del lazo cerrado.
modo_control	char	Flag de estado de control: 1-> Lazo cerrado 0-> Lazo abierto
dantes	float	Ciclo de trabajo anterior
deadband	int	Tiempo muerto
offset	float	Offset introducido mediante AD620
count	int	Aparece dentro del timer, cuenta hasta 200 lo cual equivale a un periodo de señal.
j	int	Aparece dentro del timer, cuenta hasta 100 para realizar la media de los 100 valores almacenados.

4.3.2 Variables globales en modulación por desplazamiento de fase

Tabla 7

Nombre	Tipo	Función
fVo	Float	Almacena el valor en cada lectura del ADC.
Seleccionprinc Selecamp selecdef	char	Variables de selección, almacenan la selección en cada menú a través de la UART.
duty	float	Ciclo de trabajo de la señal PWM.
A	float	'A' se asemeja a la modulación en amplitud teórica.
offset	float	Almacena la suma de 100 lecturas del ADC y el promedio de estas.
offsetfin	float	Almacena el cuadrado de la

		variable offset.
count	int	Cuenta hasta 100 para realizar la media de 100 valores.
sen[20]	float	Array constante, almacena 20 muestras equidistantes de la función seno para modular.
i	int	Cuenta hasta 20 y sirve para recorrer el array Sen[20].
tensionfinal	float	Valor de la amplitud de la onda a la salida del inversor.
Aantes	float	Valor de A en el ciclo anterior.
errorahora	float	Diferencia entre tensionfinal y tensionref.
errorganancia	float	Errorahora multiplicado por la ganancia del lazo cerrado.
k	float	Ganancia del lazo cerrado.
modo_control	char	Flag de estado de control: 1-> Lazo cerrado 0-> Lazo abierto
tensionref	float	Tensión que se desea alcanzar en lazo cerrado.
tensione	float	Acumula la suma del cuadrado de 100 muestras tomadas para realizar la media, es el cuadrado de la tensión eficaz de la onda.
tensione	float	Tensione-offsetfin

4.3.3 Set Up

A continuación se hará una descripción detallada de la configuración de cada uno de los periféricos empleados para el diseño del sistema.

Para empezar es necesario configurar la frecuencia del reloj del sistema, en el presente trabajo se han utilizado dos montajes distintos para dos modos distintos de modulación.

En la modulación por desplazamiento de fase se ha seleccionado una frecuencia de reloj de 2.86 MHz, fue necesario reducir hasta dicho valor la frecuencia para conseguir hacer conmutar los transistores a una frecuencia de 50 Hz sin que el reloj del sistema desbordase, 50Hz es la frecuencia a la que se desea obtener la tensión de salida del convertidor.

```
g_ui32SysClock = SysCtlClockFreqSet((SYSCTL_XTAL_25MHZ |
                                     SYSCTL_OSC_MAIN |
                                     SYSCTL_USE_PLL |
                                     SYSCTL_CFG_VCO_320), 2860000);
```

Por otro lado, para la modulación en modo PWM Bipolar la frecuencia de reloj seleccionada es de 120 MHz.

4.3.3.1 Estrategia de generación para las señales PWM en modulación por desplazamiento de fase

La estrategia de generación de las señales PWM empleada en el presente trabajo para la modulación por desplazamiento de fase es originaria de [6], solo funcionará para ciclos de trabajo del 50%, en este caso el ciclo de trabajo siempre será igual al 50%, lo único que variará será el desfase entre señales.

Para llevar a cabo dicha estrategia es necesario configurar un generador de PWM en modo UP/DOWN (PWM0) y otro generador en modo DOWN o UP, cualquiera de los dos modos serviría (PWM1).

El primer generador (PWM0) se configura a una frecuencia igual a la frecuencia de conmutación, de tal forma que cuando el valor del contador sea igual al de su comparador, la señal cambiara de nivel generándose una señal de ciclo de trabajo 50% a la frecuencia deseada.

El segundo generador (PWM1) se configurará a una frecuencia igual al doble de la frecuencia de conmutación, de tal forma que cuando el valor de cuenta alcance el valor de su comparador, la señal cambiará de nivel. El comparador del segundo generador se configurará con el valor deseado para el desfase, es decir, si deseamos un desfase del 30% el valor del comparador será igual a 0.3 por la frecuencia con la que se haya programado el generador PWM1.

La estrategia de generación descrita anteriormente se presenta en la **Figura 27** original de [4].

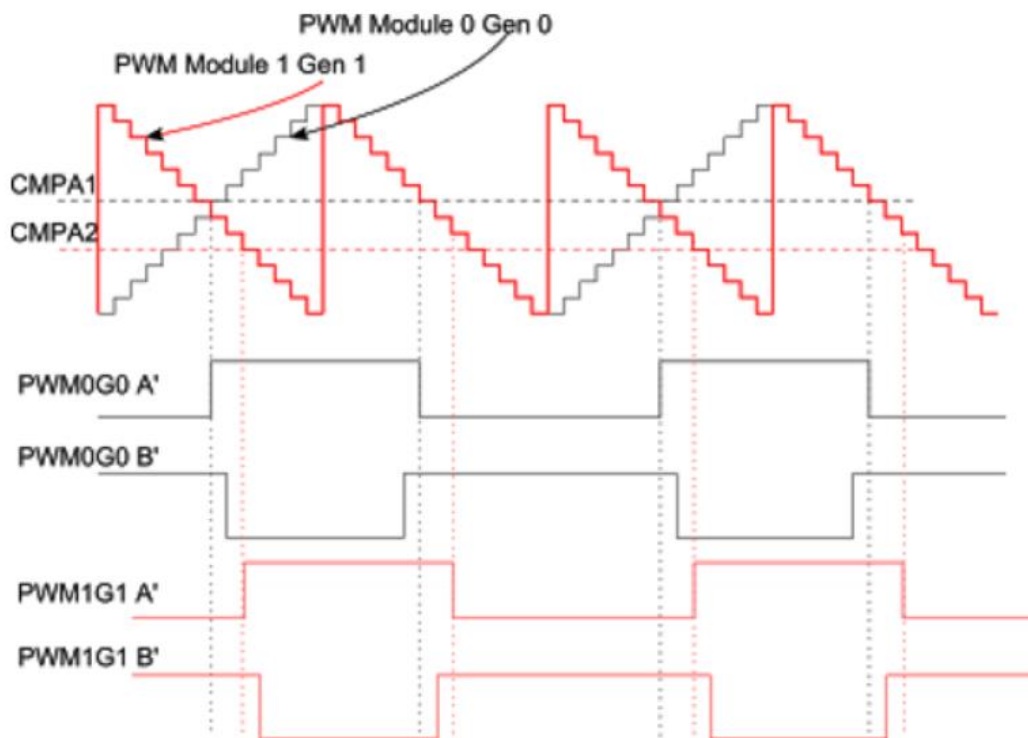


Figura 27

La correspondencia entre las señales generadas de dicha estrategia y los transistores del puente inversor es la siguiente:

PWM0G0 A'	T2
PWM0G0 B'	T4
PWM1G1 A'	T3
PWM1G1 B'	T1

Una vez explicada la estrategia de generación de las señales PWM se explicará de forma detallada el código utilizado para su configuración, dicho código ha sido extraído de las librerías **sysctl.h**, **pwm.h**, **gpio.h**.

En primer lugar es necesario sincronizar el periférico con el reloj que gobierna el sistema:

```
SysCtlPWMClockSet(SYSCTL_PWMDIV_1);
```

A continuación se habilitan los periféricos que van a ser utilizados:

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_PWM0);
```

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
```

Se indica en que pines del puerto F se generará la señal y su complementaria y los configuro como salida PWM:

```
GPIOinConfigure(GPIO_PF0_M0PWM0); GPIOinConfigure(GPIO_PF1_M0PWM0);
```

```
GPIOinTypePWM(GPIO_PORTF_BASE,GPIO_PIN_0);
```

```
GPIOinTypePWM(GPIO_PORTF_BASE,GPIO_PIN_1);
```

Se configura el modo de funcionamiento del generador PWM0:

```
PWMGenConfigure(PWM0_BASE,PWM_GEN_0,PWM_GEN_MODE_UP_DOWN|PWM_GEN_MODE_NO_SYNC);
```

Se configura el contador a frecuencia de conmutación:

```
PWMGenPeriodSet(PWM0_BASE,PWM_GEN_0,262000);
```

Se asigna un ciclo de trabajo del 50% en el caso del primer generador y del desfase en el caso del segundo:

```
PWMPulseWidthSet(PWM0_BASE,PWM_OUT_0,PWMGenPeriodGet(PWM0_BASE,PWM_GEN_0)*0.5);
```

Se da valor al tiempo muerto, cuya función será descrita en el apartado 4.3.3.3:

```
PWMDeadBandEnable(PWM0_BASE,PWM_GEN_0, deadband, deadband);
```

Se impone la condición de que las señales generadas en cada pin sean opuestas:

```
PWMOutputState(PWM0_BASE,PWM_OUT_1_BIT|PWM_OUT_0_BIT, true);
```

y finalmente se habilita el generador:

```
PWMGenEnable(PWM0_BASE,PWM_GEN_0);
```

En las siguientes imágenes se muestra el código que se ha utilizado para programar ambos generadores:

```
// PWM0.  
//  
SysCtlPWMClockSet(SYSCTL_PWMDIV_1);  
SysCtlPeripheralEnable(SYSCTL_PERIPH_PWM0);  
  
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);  
GPIOPinConfigure(GPIO_PF0_M0PWM0);  
GPIOPinConfigure(GPIO_PF1_M0PWM1);  
GPIOinTypePWM(GPIO_PORTF_BASE,GPIO_PIN_0);  
GPIOinTypePWM(GPIO_PORTF_BASE,GPIO_PIN_1);  
PWMGenConfigure(PWM0_BASE,PWM_GEN_0,PWM_GEN_MODE_UP_DOWN|PWM_GEN_MODE_NO_SYNC);  
PWMGenPeriodSet(PWM0_BASE, PWM_GEN_0, 262000);  
PWMPulseWidthSet(PWM0_BASE,PWM_OUT_0,PWMGenPeriodGet(PWM0_BASE, PWM_GEN_0)*50/100);  
PWMDeadBandEnable(PWM0_BASE,PWM_GEN_0, deadband, deadband);  
PWMOutputState(PWM0_BASE,PWM_OUT_1_BIT | PWM_OUT_0_BIT, true);  
PWMGenEnable(PWM0_BASE, PWM_GEN_0);
```

```
//PWM1
```

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
GPIOPinConfigure(GPIO_PF2_M0PWM2);
GPIOPinConfigure(GPIO_PF3_M0PWM3);
GPIOPinTypePWM(GPIO_PORTF_BASE,GPIO_PIN_2);
GPIOPinTypePWM(GPIO_PORTF_BASE,GPIO_PIN_3);
PWMGenConfigure(PWM0_BASE,PWM_GEN_1,PWM_GEN_MODE_DOWN|PWM_GEN_MODE_NO_SYNC);
PWMGenPeriodSet(PWM0_BASE, PWM_GEN_1, 131000);
PWMPulseWidthSet(PWM0_BASE,PWM_OUT_2,PWMGenPeriodGet(PWM0_BASE, PWM_GEN_1)*30/100);
PWMDeadBandEnable(PWM0_BASE,PWM_GEN_1,deadband,deadband);
PWMOutputState(PWM0_BASE,PWM_OUT_3_BIT | PWM_OUT_2_BIT, true);
HWREG(0x40028000 + 0x0A0)=0x440;
PWMGenEnable(PWM0_BASE, PWM_GEN_1);
```

La **Figura 28** muestra las señales que gobiernan el disparo de los cuatro transistores.

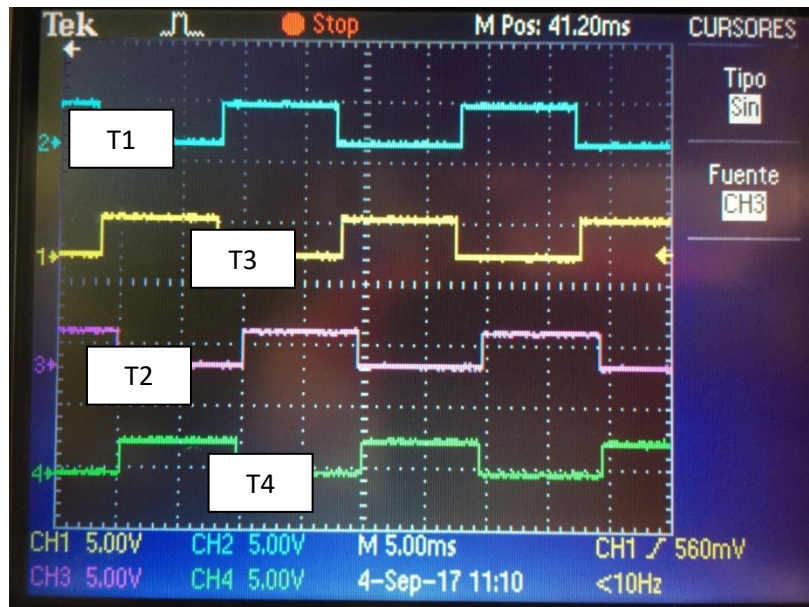


Figura 28

4.3.3.2 Estrategia de generación para las señales PWM en modulación PWM bipolar

En la modulación por PWM bipolar solo es necesario el uso de un generador de señales PWM. El resultado es una señal PWM en el pin PF0 y su complementaria en el pin PF1, sin embargo, para conseguir el efecto mostrado en la **Figura 29** es necesario variar este ciclo de trabajo en un periodo de 0.02 s (50 Hz) haciendo una aproximación a la función que llevan a cabo la señal moduladora y la señal portadora en la estrategia de generación típica de esta modulación.

Para variar el ciclo de trabajo se decidió almacenar en un array denominado sen[20], veinte valores obtenidos al muestrear una función seno de amplitud la unidad. En cada interrupción del timer a 1000 Hz de frecuencia (0.001 s) asigno un valor distinto al ciclo de trabajo (d) que se obtendrá recorriendo el vector sen[20], por tanto, se tendrá una señal PWM que varía su ciclo de trabajo a una frecuencia de 1000 Hz y que además varía 20 veces, repitiéndose esta

variación cada T_{red} siendo $T_{red} = T_{timer} * 20 = 0.02 \text{ s}$ que equivale a una frecuencia de 50 Hz (frecuencia de red) que es el valor de frecuencia que queremos ver para la señal de salida del convertidor inversor.

Al final se obtienen dos señales, la señal del pin PF0 que se correspondería con la señal morada en la **Figura 29** y la señal del pin PF1 que se corresponde con la señal amarilla en la **Figura 29** dando lugar en su diferenciación a la tensión V_{AB} que mostrábamos en el **apartado 2.3.2**. Además en la **Figura 30** se muestra una medida con el osciloscopio de estas mismas señales.

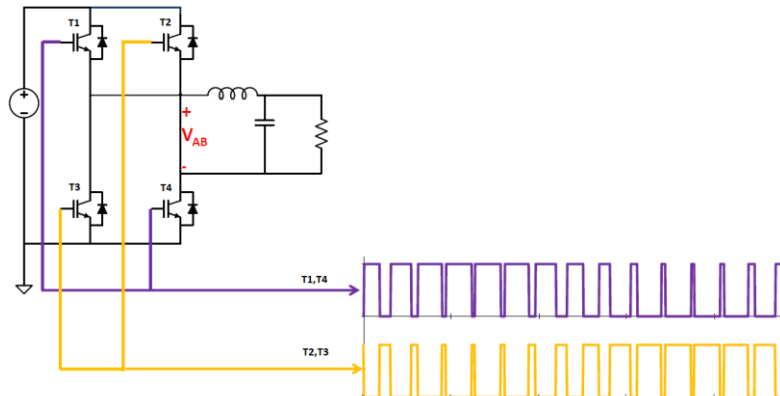


Figura 29

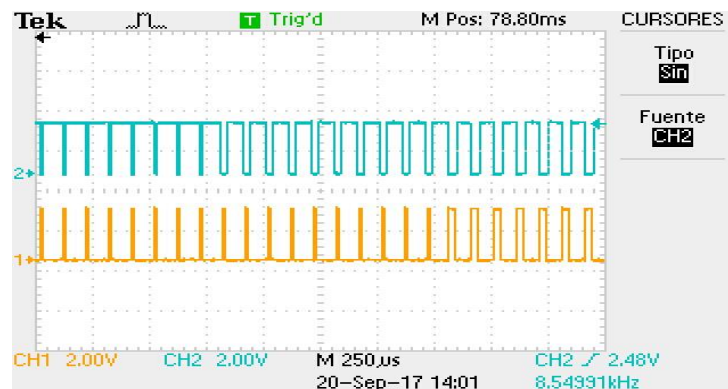


Figura 30

En la **figura 31** se muestra una gráfica en la que aparecen los valores que contiene la variable $\sin[20]$, como se puede apreciar, corresponde con una función seno con una amplitud de 0.5, y una frecuencia de 50 Hz. Además se ha añadido un offset de 0.5 para que el ciclo de trabajo se mueva en un rango de valores entre 0 y 1.

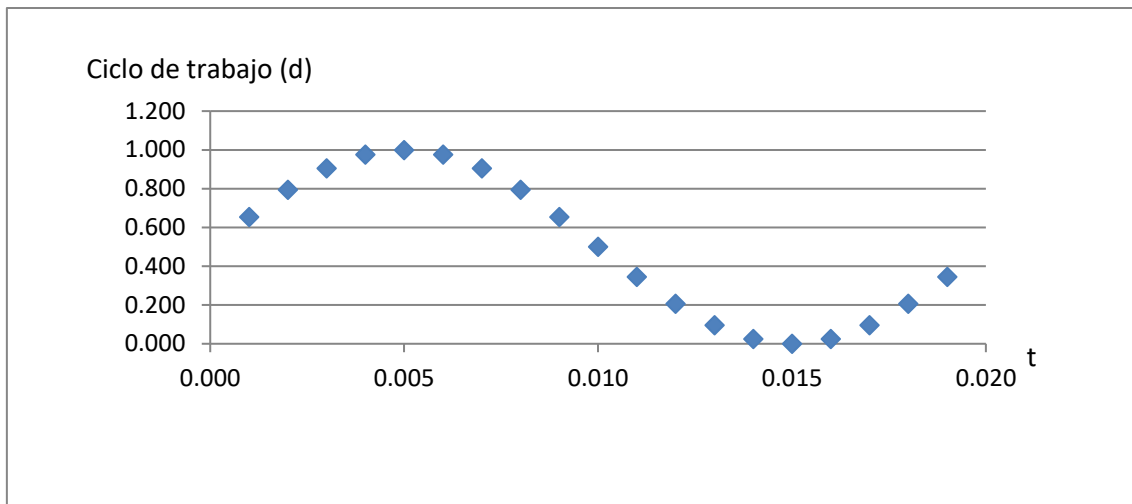


Figura 31

4.3.3.3 Tiempos Muertos

Los tiempos muertos o deadband, son tiempos de retraso introducidos en el sistema mediante la programación del micro controlador Tiva C Series EK-TM4C1294XL, y su finalidad es conseguir que la conmutación de transistores de una misma rama se lleve a cabo correctamente y no coincidan a la vez ambos cerrados o abiertos. Este es el caso por ejemplo de los transistores 1 y 3, o de los transistores 2 y 4.

A continuación, en la **Figura 32**, se presenta una medida mediante el osciloscopio de las señales que gobiernan los mosfet, y en ella se puede apreciar que, cuando la señal que gobierna el transistor 1 cambia a nivel bajo, la señal que gobierna el transistor 3 tarda un tiempo en ponerse a nivel alto.

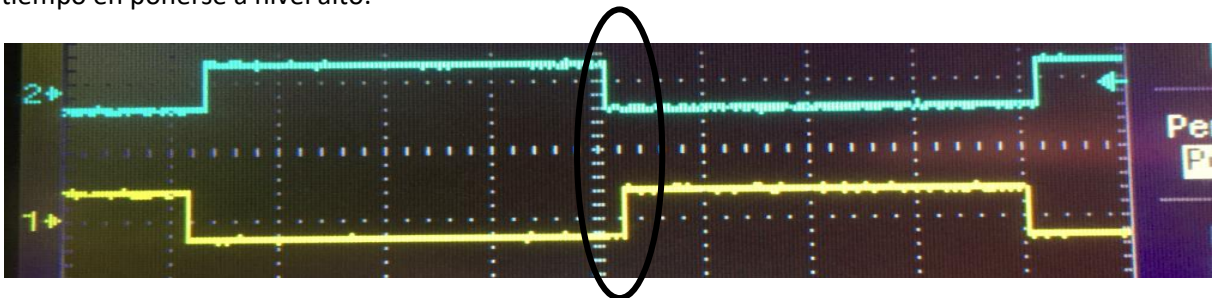


Figura 32

4.3.3.4 Timer

Las funciones utilizadas para programar el timer periódico han sido obtenidas de la biblioteca rom.h.

En primer lugar se habilita el periférico:

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER1);
```

Se configura como un timer periódico:

TimerConfigure(TIMER1_BASE,TIMER_CFG_PERIODIC);

Se introduce el valor de cuenta requerido, que en nuestro caso será de 100 us en la modulación por desplazamiento de fase y de 1000 us en la modulación bipolar.

TimerLoadSet(TIMER0_BASE,TIMER_A, g_ui32SysClock/10000);

La siguiente función hace que se habilite una lectura del ADC cada vez que se produzca una interrupción del timer:

TimerControlTrigger(TIMER1_BASE,TIMER_B,true);

Por último se habilitan las interrupciones del timer y se habilita la cuenta:

IntEnable(INT_TIMER0A);

TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT);

TimerEnable(TIMER0_BASE, TIMER_A);

En la siguiente imagen, se muestra el código empleado.

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0);
TimerConfigure(TIMER0_BASE, TIMER_CFG_PERIODIC);
TimerLoadSet(TIMER0_BASE, TIMER_A, g_ui32SysClock/10000);
TimerControlTrigger(TIMER0_BASE,TIMER_A,true);
IntEnable(INT_TIMER0A);
TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT);
TimerEnable(TIMER0_BASE, TIMER_A);
```

4.3.3.5 ADC

Las funciones empleadas para la programación del convertidor analógico-digital fueron obtenidas de las bibliotecas **sysctl.h**, **gpio.h** y **adc.h**.

Para comenzar, se habilitan los periféricos que se van a utilizar, en nuestro caso, el ADC0 y el pin a través del cual leeremos la señal analógica.

SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);

SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);

Se configura el pin 3 del puerto E como entrada para el ADC:

GPIOinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_3);

A continuación se habilita la secuencia 3 con el Processor Trigger para realizar una muestra cada vez que sea necesario:

ADCSequenceConfigure(ADC0_BASE, 3, ADC_TRIGGER_TIMER, 0);

Se configura el paso 0 en la secuencia 3, canal 0 en el modo single-ended y la activación del flag cuando la conversión haya finalizado:

ADCSequenceStepConfigure(ADC0_BASE, 3, 0, ADC_CTL_CH0|ADC_CTL_IE|ADC_CTL_END);

Se habilita la secuencia 3:

ADCSequenceEnable(ADC0_BASE, 3);

Se limpia el estado del flag de interrupción antes de comenzar con la lectura:

ADCIntClear(ADC0_BASE, 3);

El código empleado para programar el ADC es el representado en la siguiente imagen.

```
// ADC

SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
GPIOinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_3);
ADCSequenceConfigure(ADC0_BASE, 3, ADC_TRIGGER_TIMER, 0);
ADCSequenceStepConfigure(ADC0_BASE, 3, 0, ADC_CTL_CH0 | ADC_CTL_IE | ADC_CTL_END);
ADCSequenceEnable(ADC0_BASE, 3);
ADCIntClear(ADC0_BASE, 3);
```

4.3.3.6 UART (Protocolo comunicación serie)

Las funciones empleadas para programar la UART han sido extraídas de las bibliotecas **rom.h** y **uart.h**.

Para la programación ha sido necesario el uso de la función **ConfigureUart()**.

En primer lugar, y como en casi todos los apartados anteriores, se habilita la UART0 y el puerto A, de donde se utilizarán el pin 0 y el 1 como vía de transmisión y recepción de datos:

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
```

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
```

```
GPIOInConfigure(GPIO_PA0_U0RX);
```

```
GPIOInConfigure(GPIO_PA1_U0TX);
```

```
GPIOInTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);
```

También es necesario habilitar las interrupciones que se producen cuando se finaliza el envío o la recepción de datos:

```
UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT);
```

Y finalmente se selecciona una velocidad de transmisión de datos en bits/segundo:

```
UARTStdioConfig(0, 115200, g_ui32SysClock);
```

En la figura se muestra el código empleado.

```
void  
ConfigureUART(void)  
{  
  
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);  
  
    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);  
  
    GPIOPinConfigure(GPIO_PA0_U0RX);  
    GPIOPinConfigure(GPIO_PA1_U0TX);  
    GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);  
  
    UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT);  
  
    UARTStdioConfig(0, 115200, g_ui32SysClock);  
}
```

4.3.4 Diagramas de flujo y explicación de las funciones principales

Para el caso de la modulación por desplazamiento de fase el programa principal, es decir, el main, comienza con la declaración de variables, seguidamente se lleva a cabo el Set Up de cada uno de los periféricos que se van a utilizar. Este proceso se lleva a cabo una vez por cada ejecución del programa. Después del Set Up, el programa llega a un bucle infinito donde se presenta la función MenuPrincipal (), la cual será descrita con mayor detalle posteriormente.

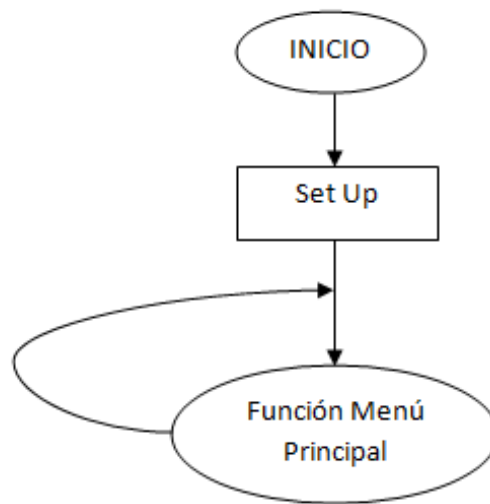


Figura 33

En la función MenúPrincipal () cuyo flujograma se presenta en la **Figura 35** se presentan cuatro posibles opciones como muestra en la **Figura 34**.

```
BIENVENIDO AL SISTEMA
1.Lazo Abierto
2.Modificar Deadband
3. Ver propiedades actuales del convertidor
4. Lazo Cerrado
```

Figura 34

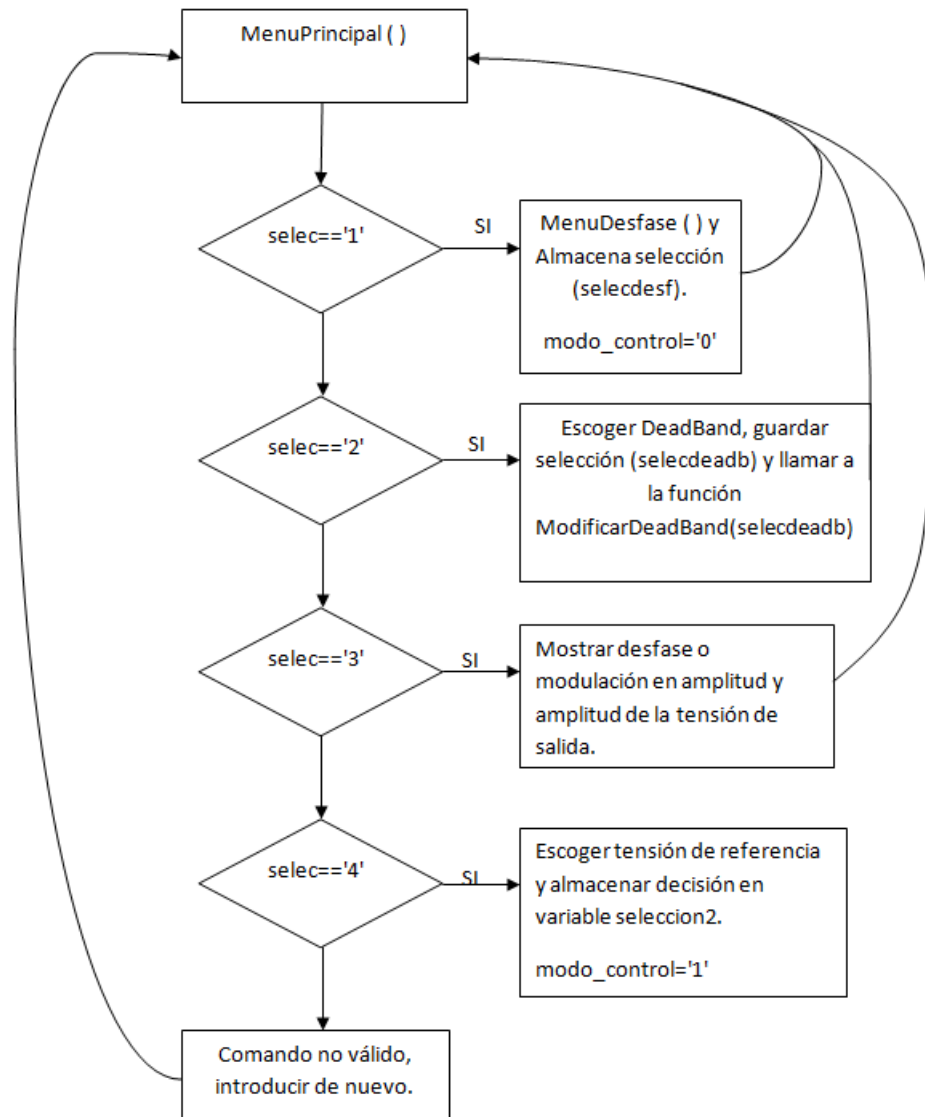


Figura 35

1. Lazo Abierto: Mediante esta opción se podrá interactuar con el convertidor modificando el desfase de las señales PWM y viendo como repercute sobre la tensión de salida. En esta opción se ejecuta una función que presenta el menú con los diferentes desfases:

```
Seleccione un desfase:
1. Desfase 10 por ciento
2. Desfase 20 por ciento
3. Desfase 30 por ciento
4. Desfase 40 por ciento
5. Desfase 50 por ciento
6. Desfase 60 por ciento
7. Desfase 70 por ciento
```

Figura 36

El flujo grama de dicha función es sencillo puesto que únicamente imprime caracteres a través de la UART:

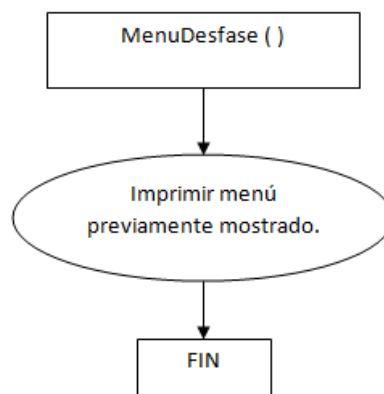


Figura 37

Al seleccionar uno de los desfases, esta selección queda almacenada en la variable 'selecdef' que posteriormente se pasará como parámetro a la función `LazoAbierto (selecdef)` que será la encargada de fijar en el generador PWM el desfase seleccionado. El diagrama de flujo de dicha función es el siguiente:

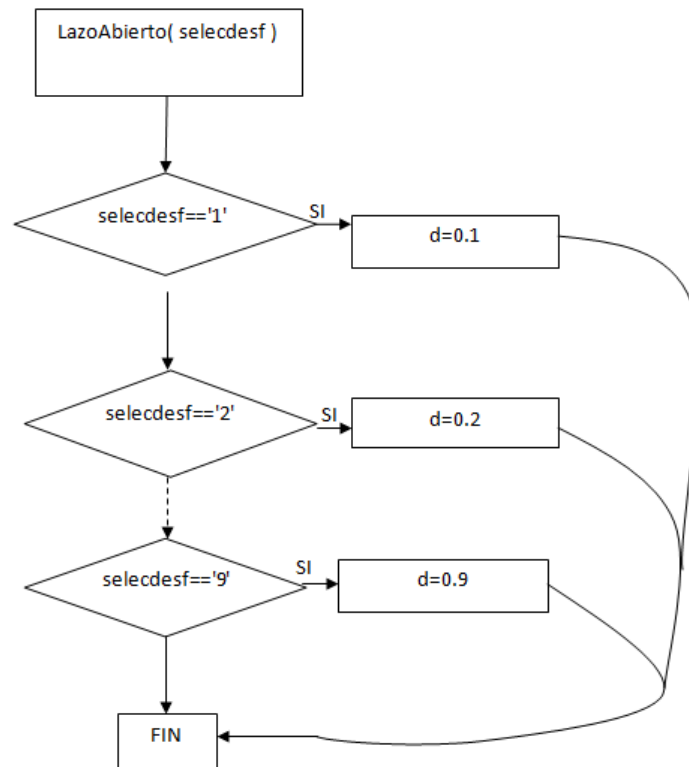


Figura 38

La función LazoAbierto (selecdef) es llamada en la interrupción del timer para que se produzca una actualización continua, pero solo actuará cuando el flag 'modo_control' tome el valor '0', valor que tomará cuando en el menú principal escojamos la opción de Lazo Abierto, y que dejará de valer '0' cuando escojamos la opción Lazo Cerrado. El uso del flag 'modo_control' es necesario para evitar que la actualización del ciclo de trabajo en ambos modos de control no se superponga.

2. Modificar Deadband: Con esta opción podremos modificar el tiempo muerto necesario para que se produzca una correcta conmutación, cuya importancia fue explicada en el **apartado 4.3.3.3**.

A través de la pantalla se mostrará lo siguiente:

```

MODIFICAR DEADBAND
1.10 Por ciento
2.15 Por ciento
3. 20 Por ciento
4. 25 Por ciento
  
```

Figura 39

La opción elegida se almacenará en la variable 'selecdeadb' y se pasará como parámetro de la función ModificarDeadBand (selecdeadb) cuyo diagrama de flujo es el siguiente:

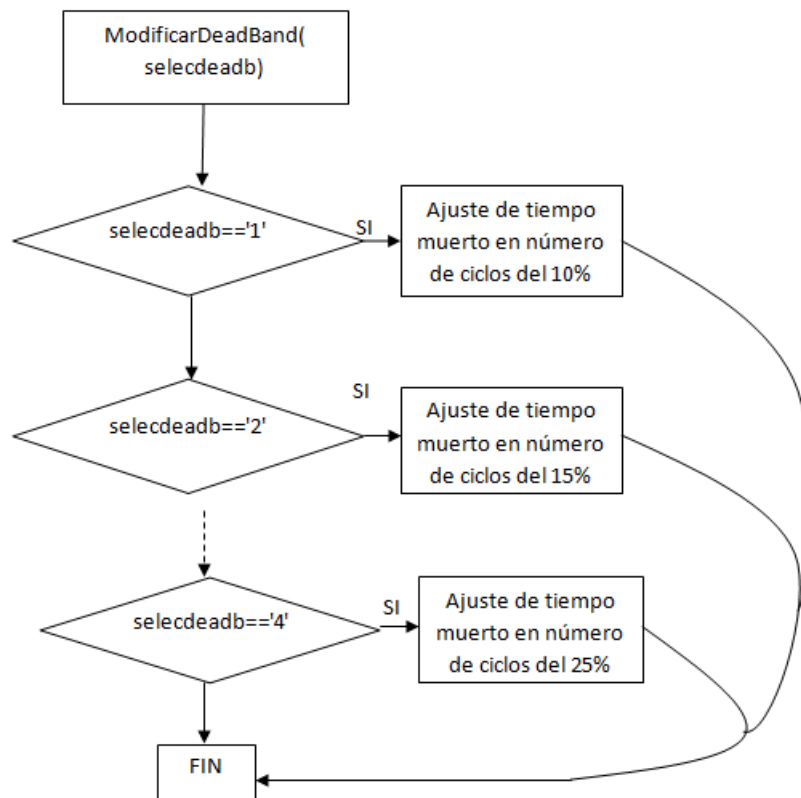


Figura 40

3. Ver propiedades del convertidor: Se podrán visualizar por pantalla las características que tiene el convertidor en ese preciso instante, así como el desfase existente, la modulación en amplitud en el caso de PWM bipolar, y la amplitud de la tensión de salida.

Para mostrar dichas características se emplea la siguiente sentencia:

```
UARTprintf("Desfase = %3d.%03d\n", int32IntegerPartD, int32FractionPartD);
```

Donde las variables 'int32IntegerPartD' y 'int32FractionPartD' se encargan de mostrar la parte entera y decimal de un número binario empleando el siguiente algoritmo:

En primer lugar se convierte la variable 'd' de tipo float a una variable entera de 32 bits almacenando la parte entera de dicha variable.

```
int32IntegerPartD=(int32_t)d;
```

En segundo lugar se lleva a cabo el mismo proceso, pero en este caso se multiplica la variable por 1000 para almacenar también la parte decimal.

```
int32FractionPartD=(int32_t)(d*1000.0f);
```

Por último se resta a la variable que contiene el número completo con decimales la variable que contiene solo el número entero quedándonos así únicamente con la parte decimal.

```
int32FractionPartD=int32FractionPartD-(int32IntegerPartD*1000);
```

Esta última sentencia tiene como finalidad convertir a positivos aquellos números que resulten negativos tras la diferencia aplicada en la sentencia anterior.

```
if(int32FractionPartD<0){  
    int32FractionPartD*= -1;}
```

Este mismo algoritmo es repetido cada vez que se desea mostrar una variable con decimales por pantalla.

4. Lazo Cerrado: Con esta opción se accede al modo de control en Lazo Cerrado, se podrá seleccionar una tensión de referencia, o tensión a alcanzar, y el convertidor de forma automática regulará el desfase o la modulación en amplitud para alcanzar dicha tensión, siempre y cuando esta se encuentre en el rango de valores alcanzables.

El flujo grama de dicha función es el siguiente:

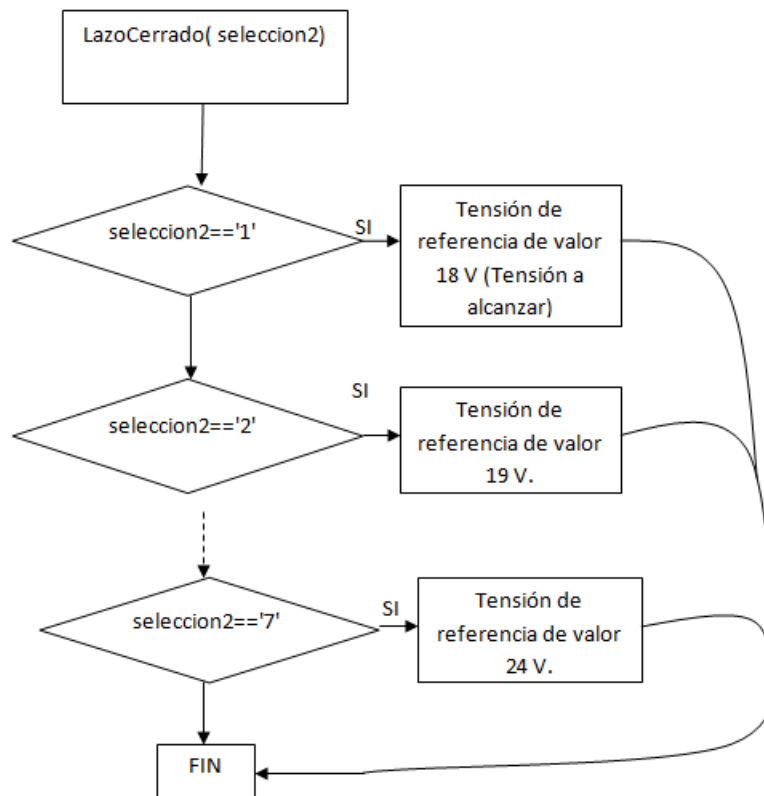


Figura 41

Además, en esta opción se ejecuta una función cuyo objetivo es mostrar un menú con los diferentes valores de tensión de referencia. Su flujograma es similar al empleado en la opción 1 para mostrar las opciones de desfase existentes.

Para finalizar, en el caso de la modulación PWM Bipolar, las funciones empleadas se asemejan a las descritas previamente, pero son adaptadas para esta modulación de tal forma que en la función ModificarAmplitud(char seleccion) se modifica la variable A que se corresponde con la modulación en amplitud en vez de el desfase (d). En la **figura 42** se muestra el flujograma de la función previamente nombrada.

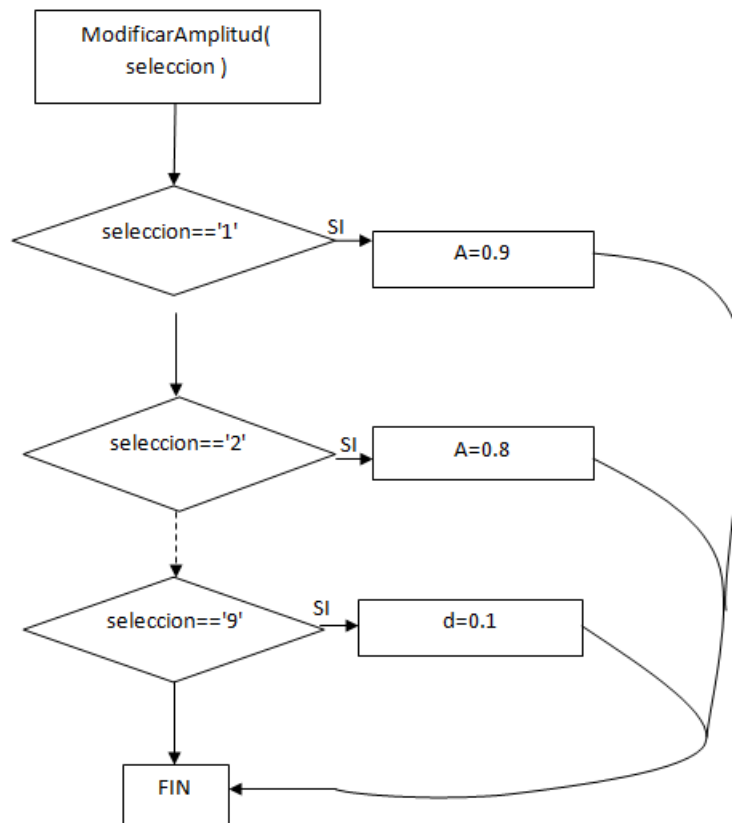


Figura 42

PRUEBAS EXPERIMENTALES

En el capítulo 5, se presentan una serie de tablas que reflejan medidas reales en contraste con valores teóricos tanto en lazo cerrado como en lazo abierto y para ambas modulaciones.

5.1 Modulación por desplazamiento de fase

5.1.1 Lazo Abierto

En primer lugar, en la tabla siguiente se muestra la variación de la tensión para valores de desfase comprendidos en el intervalo [0,1].

Tabla 8

Desfase (d)	Amplitud de la tensión (V)	Amplitud teórica (V)
0.99	17.6	19.91
0.9	19.2	18.2
0.8	20.8	21.6
0.7	22.4	26
0.6	23.6	29
0.5	24.4	30
0.55	24.2	29.85
0.4	24.1	29.0
0.3	23.2	25.99
0.2	21.2	21.6
0.1	19.6	18.2
0.01	18	20

En el gráfico se puede comprobar que para un ciclo de trabajo de 0.01 el valor de la amplitud es mínimo, a medida que aumentamos el desfase, la amplitud aumenta hasta llegar a 0.5, a partir de entonces comienza a disminuir de nuevo. Esto se debe a que V_{AB} puede alcanzar el mismo valor para un cierto desfase comprendido entre 0° y 180° que para un desfase comprendido entre 180° y 360° lo único que se modificaría sería la polaridad, pero al ser una componente alterna no influye en nada. Por tanto las unidades de d serían en tanto por uno de un semiperiodo (180°). Como se observa en la **figura 43** el comportamiento que tiene el sistema en la realidad es similar al que tiene el sistema teórico aunque el valor de amplitud cambia ligeramente. Esto puede ser debido a que la presencia de armónicos en el caso teórico no sea exactamente idéntica a la del caso real, además puede ser debido también a la existencia de pérdidas en el circuito.

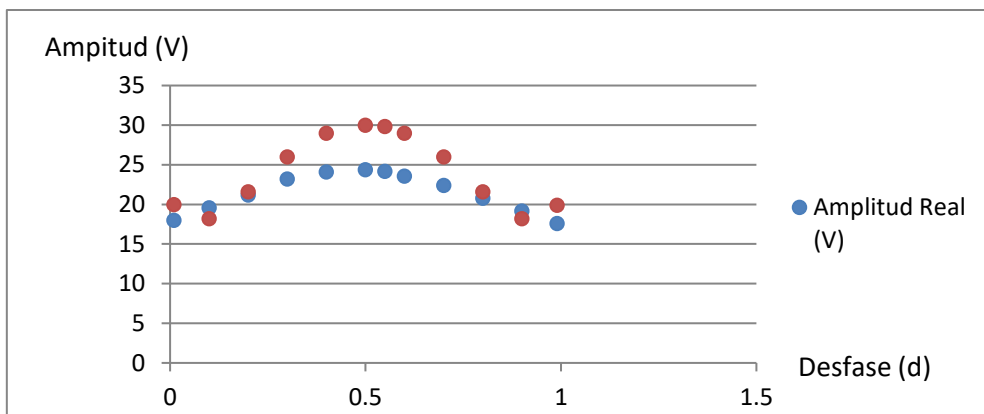


Figura 43

5.1.2 Lazo Cerrado

En este apartado, el modo de control es lazo cerrado, por lo que la variable modo_control será igual a 1. Se muestran dos tablas, en la **Tabla 9** fijaré las distintas tensiones de referencia ofrecidas por el menú y se contrastará con el valor real obtenido.

Tabla 9

$V_{\text{referencia}}$ (V)	V_{real} (V)
18	18.17
19	19.09
20	20.04
21	21.01
22	22.02
23	23.09
24	23.95

Como se observa, el valor real obtenido es muy próximo al valor esperado, luego el control en lazo cerrado funciona correctamente, pero es necesario verificar si el regulador integrador responde también ante perturbaciones en la tensión de entrada (V_g). Para ello se ha fijado la tensión de referencia en 17 V y se tomará una medida de amplitud de la tensión de salida para diferentes valores de V_g .

Tabla 10

V_g (V)	V_{real} (V)
11	17.1
11.5	17.02
12	17.06
12.5	17.01
13	17.02
13.5	17.05
14	17.05

14.5	17.07
15	17.04

Finalmente se corrobora que realmente el regulador integrador funciona.

5.2 Modulación PWM Bipolar

5.2.1 Lazo Abierto

En la tabla mostrada a continuación se muestra la amplitud de la tensión de salida del convertidor inversor para diferentes valores de A (modulación en amplitud), además se contrasta con el valor teórico obtenido a partir de simulaciones en PWIM:

Tabla 11

Modulación en amplitud (A)	Amplitud de la tensión (V)	Amplitud teórica (V)
0.1	1.6	1.6
0.2	3.1	3.4
0.3	4.9	5.1
0.4	6.8	7
0.5	8.6	8.6
0.6	10.4	10.3
0.7	12.2	12
0.8	14.2	13.6
0.9	16.2	15.2
1	17.6	16.9

Como puede verse en la tabla, existe una gran similitud entre los resultados teóricos y los reales, por lo que puede concretarse que la modulación es bastante aproximada al método teórico consistente en comparar señal moduladora y portadora.

A continuación, en la **figura 44**, se muestra un grafico en el que se compara la amplitud teórica con la amplitud real obtenida para valores de modulación en amplitud entre 0 y 1.

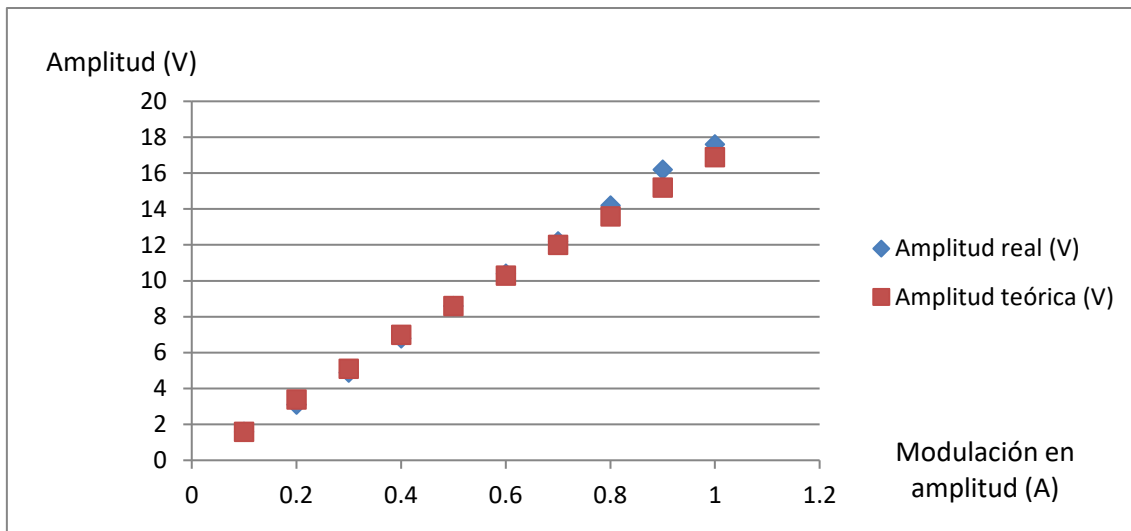


Figura 44

5.2.2 Lazo Cerrado

Para el modo de control en lazo cerrado se han fijado los diferentes valores de tensión de referencia ofrecidos en el menú para comprobar si realmente la amplitud de la tensión de salida se estabiliza entorno a ese valor sin generar oscilaciones, los resultados se muestran en la siguiente tabla:

Tabla 12

$V_{\text{referencia}}$ (V)	V_{real} (V)
4	3.99
6	6.02
8	8.03
10	10.1
12	11.98
14	14.04
16	15.97

Para demostrar que realmente el regulador responde como se espera, se ha hecho variar la tensión de entrada al convertidor inversor (V_g) obteniéndose los siguientes resultados:

Tabla 13

V_g (V)	$V_{\text{referencia}}$ (V)	V_{real} (V)
9	9	9.07
10	9	8.99
11	9	9.07
12	9	9.05
13	9	9.06
14	9	9.03
15	9	9.01

EJERCICIO PROPUESTO A ALUMNOS DE LA ASIGNATURA ELECTRÓNICA DE POTENCIA

En el capítulo 6 se van a proponer una serie de ejercicios para que lleven a cabo los alumnos que cursen la asignatura de electrónica de potencia u otras relacionadas, ya que la principal finalidad de el presente trabajo es desarrollar una interfaz intuitiva entre alumno y convertidor, de forma que estos puedan interactuar con el convertidor y puedan entender de una forma más ilustrativa su funcionamiento. Los objetivos fundamentales de la prácticas son:

- 1) El alumno aprenderá las diferencias fundamentales entre modular por desplazamiento de fase y por PWM Bipolar.
- 2) Que el alumno sepa relacionar la calidad de la señal con el tipo de modulación empleada basándose en la distorsión armónica total (DAT).
- 3) Que el alumno adquiera una experiencia práctica en el control de convertidores CC/CA.

6.1 Cálculos teóricos previos

A continuación se proponen una serie de cálculos previos que se recomienda resolver antes de la realización de la práctica.

6.1.1 Cálculos para modulación por desplazamiento de fase

Teniendo en cuenta los valores que se muestran en la siguiente tabla, calcular el valor de amplitud de los tres primeros armónicos antes y después de la ganancia o atenuación que sufrirán debido al filtro LC, así como el valor de distorsión armónica total.

Tabla 14

L	520 mH
C	3.3 uF
V_G	15 V
α	45°
R	1200 Ω

L= Valor de la inductancia del filtro

C=Valor de la capacitancia del filtro

R=Valor resistivo del filtro

V_G = Tensión continua aplicada a la entrada del puente inversor

α = Desfase

Tabla 15

N	Frecuencia (Hz)	V_{AB} (V)	Ganancia del filtro	V_o (V)
1				
3				
5				

DAT =

Repita los cálculos sustituyendo la capacidad del condensador del filtro por 10 μ F.

Tabla 16

N	Frecuencia (Hz)	V_{AB} (V)	Ganancia del filtro	V_o (V)
1				
3				
5				

DAT =

¿Qué efecto puede apreciarse en el cálculo de la distorsión armónica total?

Se aprecia que, la DAT para el filtro con $C=3.3 \mu$ F es mucho más elevada que la obtenida para el filtro con $C=10 \mu$ F.

¿A que es debido dicho efecto?

Este efecto es debido principalmente a que, para este tipo de modulación los armónicos aparecen a frecuencias muy cercanas a la del armónico fundamental y por este motivo resulta difícil filtrar únicamente la componente de 50 Hz.

¿Cómo repercutirá sobre la onda a la salida del filtro?

La onda a la salida del filtro presentará cierta distorsión para el caso del filtro con $C= 3.3 \mu$ F debido a la presencia de componentes indeseables que no logran ser atenuadas.

6.1.2 Cálculos para modulación por PWM bipolar

Teniendo en cuenta una modulación en amplitud $m_a = 1$, una frecuencia de la señal moduladora de 50 Hz, una frecuencia de la señal portadora de 1000 Hz y los datos que se

muestran en la tabla del apartado 6.1.1, calcule el valor de amplitud de los tres primeros armónicos así como la atenuación debida al filtro y la DAT resultante.

Tabla 17

N	Frecuencia (Hz)	$V_{AB}(V)$	Ganancia del filtro	$V_o(V)$
1				
3				
5				

DAT =

Repita los cálculos para una capacidad del condensador del filtro de 10 μF

Tabla 18

N	Frecuencia (Hz)	$V_{AB}(V)$	Ganancia del filtro	$V_o(V)$
1				
3				
5				

DAT =

Para el condensador de 3.3 μF , ¿Qué diferencia puede apreciarse entre la modulación por desplazamiento de fase y la modulación por PWM bipolar?

La distorsión armónica total es baja a diferencia de la calculada para el caso de la modulación por desplazamiento de fase a pesar de estar utilizando el mismo filtro.

¿A qué se debe tal diferencia?

La diferencia reside en que, en la modulación por PWM Bipolar, los armónicos se producen a una frecuencia cercana a la frecuencia de modulación, que en este caso tiene un valor de aproximadamente 1000 Hz, por tanto el filtro diseñado será más que suficiente para atenuar el efecto de los armónicos.

Por tanto, ¿Qué modulación considera más efectiva?

De los apartados anteriores se deduce que es más efectiva la modulación Bipolar ya que es más sencilla de filtrar y no se necesita un filtro tan restrictivo, sin embargo, para este tipo de modulación se produce ruido debido a la conmutación a alta frecuencia.

6.2 Trabajo en el laboratorio

En este apartado el alumno llevará a cabo un análisis real, el convertidor será proporcionado por el equipo de laboratorio y deberá cargar en el micro controlador Tiva C series EK-TM4C1294XL el programa que corresponda mediante Code Composer Studio. Cabe destacar que en primer lugar, se usará el montaje con condensador de 3.3 uF en el filtro LC.

Para ello será necesario ajustar previamente la alimentación en dos fuentes de tensión:

1. Canal fijo a 5 V para alimentar los drivers del puente inversor, 15 V de uno de los dos canales regulables para alimentar el puente inversor, finalmente en el canal regulable restante se seleccionaran 15 V que será la entrada del circuito inversor (V_g).
2. Canal fijo a 5 V para regular mediante un potenciómetro de 10 k Ω el offset del AD620, y de los dos canales regulables restantes se seleccionará una alimentación simétrica de $\pm 10 V$ para alimentar el circuito AD620.

Además será necesario configurar la UART para activar el interfaz con el usuario mediante el programa Tera Term:

Set-up/Terminal/Receive/Seleccionar LF y local echo.

Velocidad de transmisión de datos: Set-up/Serial Port/Baud rate/ 115200

6.2.1 Modo desplazamiento de fase

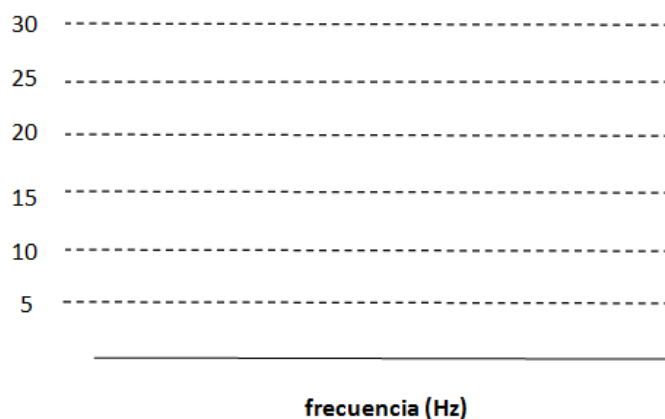
Una vez que todo esté en marcha responda a los siguientes apartados:

- a) En el menú principal, seleccione la opción Lazo Abierto, configure un desfase $d= 0.3$. A continuación seleccione la opción ver características del convertidor, compare la amplitud real con la amplitud teórica.

$$V_{Oreal} = \quad , \quad V_{Oteórica} =$$

- b) Para el mismo valor de d que en el apartado anterior, mida los armónicos de la señal de salida del convertidor inversor mediante el osciloscopio, seleccionando en el canal math la función matemática FFT. Represente el resultado en la siguiente grafica:

Amplitud de cada armónico (V)



c) Basándose en los resultados obtenidos en el apartado b), calcule la DAT.

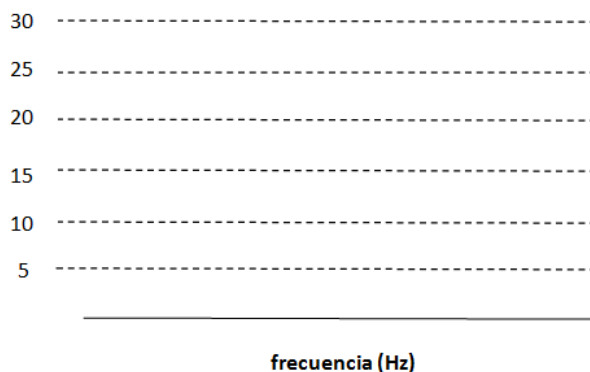
DAT=

d) En el menú principal, seleccione la opción Lazo Cerrado e indique posteriormente que tensión de referencia desea, es decir, que amplitud quieres alcanzar para la tensión a la salida del convertidor inversor. En este apartado se debe medir con el osciloscopio, mediante una sonda fija a la salida del convertidor inversor, de esta forma, al seleccionar una tensión de referencia podremos ver la acción reguladora que tiene lugar para alcanzar dicha tensión. Además podemos comprobar que el desfase entre las señales de gobierno de los transistores T1 y T2 cambia.

$V_{Oreferencia} =$, $V_{Oreal alcanzada} =$

e) Finalmente, cambie el condensador de 3.3 uF por el de 10 uF para observar el cambio que se produce sobre la tensión de salida. Realice la misma medida que en el apartado b) para este valor de C.

Amplitud de cada armónico (V)



DAT=

6.2.2 Modo PWM bipolar

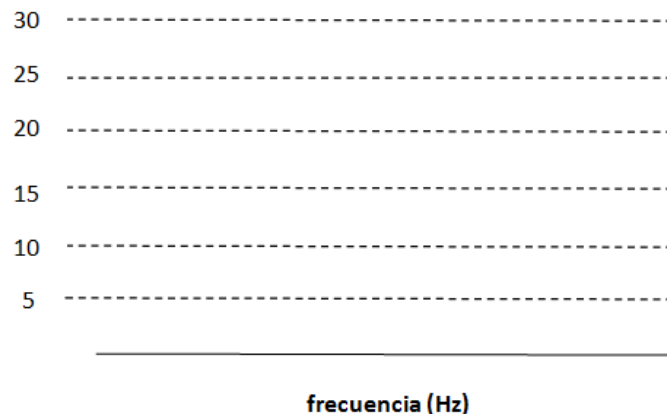
Una vez que todo esté en marcha responda a los siguientes apartados:

a) En el menú principal, seleccione la opción Lazo Abierto, configure una modulación en amplitud $A=1$. A continuación seleccione la opción ver características del convertidor, compare la amplitud real con la amplitud teórica.

$$V_{Oreal} = \quad , \quad V_{Oteórica} =$$

b) Para el mismo valor de A que en el apartado anterior, mida los armónicos de la señal de salida del convertidor inversor mediante el osciloscopio, tal y como se describió en el **apartado 6.2.1 b)**

Amplitud de cada armónico (V)



c) Basándose en los resultados obtenidos en el apartado b), calcule la DAT.

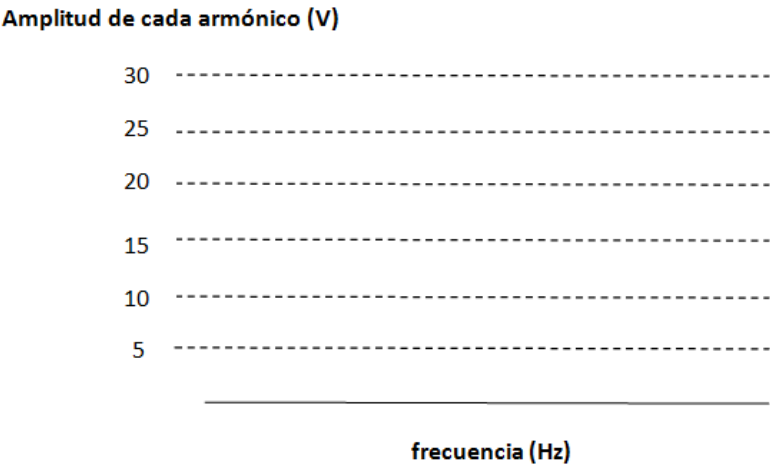
DAT=

d) En el menú principal, seleccione la opción Lazo Cerrado e indique posteriormente que tensión de referencia desea, es decir, que amplitud quieres alcanzar para la tensión a la salida del convertidor inversor. En este apartado se debe medir con el osciloscopio, mediante una sonda fija a la salida del convertidor inversor, de esta forma, al seleccionar una tensión de referencia podremos ver la acción reguladora que tiene lugar para alcanzar dicha tensión.

Además podremos ver como el valor de A cambia de forma automática para seguir la tensión de referencia escogida.

$V_{Oreferencia} = \quad , \quad V_{Oreal\ alcanzada} =$

e) Finalmente, cambie el condensador de 3.3 uF por el de 10 uF para observar el cambio que se produce sobre la tensión de salida. Realice la misma medida que en el apartado b) para este valor de C.



DAT

PRESUPUESTO Y PLANIFICACIÓN SEMANAL

7.1 Presupuesto y costes de amortización

En este capítulo se evaluará el coste de amortización de equipos, el coste de material y de personal, así como el coste de obtención de licencia para determinados software para finalmente elaborar un presupuesto aproximado para llevar a cabo el proyecto.

Para el cálculo de los costes de amortización se ha utilizado la siguiente expresión:

$$\text{Coste de amortización} = \frac{\text{Coste de equipo o licencia} * n}{\text{Vida útil de equipo o licencia}} * t$$

Donde Coste de equipo o licencia es el precio para adquirir dicho equipo o licencia, n es el número de unidades utilizadas, t es el tiempo que se ha empleado dicho equipo o licencia y Vida útil es el tiempo que puede estar operativo el equipo o la licencia.

A continuación se presenta una tabla con los equipos y software utilizados y cada uno de los datos necesarios para calcular el coste de amortización:

Tabla 19

EQUIPO/SOFTWARE	COSTE (€/Ud)	Vida útil (años)	Tiempo de uso (semanas)
Fuente de Alimentación	400,00	10	6
Osciloscopio	4000,00	10	6
Polímetro	16,00	15	3
Generador de señales	500,00	10	2
PSIM	2000,00	10	8

El cálculo detallado del presupuesto destinado a la realización del presente trabajo se detalla en la siguiente tabla:

Tabla 20

MATERIALES			
Concepto	Unidades	Precio (€/ud)	Precio total (€)
<i>Tarjeta Tiva C Series EK-TM4C1294XL</i>	1	23.77	23.77
Protoboard grande	1	3.33	3.33
Protoboard pequeña	1	1.41	1.41
Puente Inversor	1	20	20
Reactancia	1	13.46	13.46
Resistencia 1.2 KΩ	1	0.13	0.13
Resistencia 7 KΩ	2	0.33	0.66
Resistencia 290 KΩ	2	0.25	0.5
Resistencia 10 MΩ	1	0.1	0.1
Potenciómetro 100 KΩ	1	0.54	0.54
Condensador 10 uF	1	4.5	4.5
TOTAL			68.4 €

Tabla 21

AMORTIZACIÓN DE EQUIPOS			
Concepto	Unidades	Coste de amortización (€)	Precio total (€)
Fuente de alimentación	2	4.6	9.2
Osciloscopio	1	46.02	46.02
Generador de señales	1	1.91	1.91
TOTAL			57.13 €

Tabla 22

AMORTIZACIÓN DE LICENCIAS			
Concepto	Licencias (ud)	Coste de amortización (€)	Precio total (€)
PSIM 9.0.3	1	30.68	30.68
TOTAL			30.68 €

Tabla 23

RECURSOS HUMANOS			
Concepto	Número de horas	Precio (€/h)	Precio total
Diseño	40	20.00	800
Validación	90	20.00	1800

Documentación	130	15.00	1950
		TOTAL	4550 €
		PRESUPUESTO SIN I.V.A	4705.63 €
		I.V.A (21%)	988.18 €
		TOTAL	5693.81 €

7.2 Planificación semanal.

Tabla 24

Nº	Actividad	RECURSOS		JUNIO		JULIO		AGOSTO		SEPTIEMBRE	
		MATERIAL	SOFTWARE	1-15	15-30	1-15	15-31	1-15	15-31	1-15	15-30
1	Estudio previo del convertidor	Diapositivas electrónica de potencia									
1.1	Estudio del micro controlador	Programas ejemplo	CCS								
2	Montaje de convertidor modulado por desplazamiento de fase	Fuentes de alimentación, osciloscopio, PC.	PSIM, CCS								
2.1	Montaje del convertidor modulado por PWM Bipolar	Fuentes de alimentación, osciloscopio, PC.	PSIM, CCS								
3	Interfaz con el usuario (UART)	PC	CCS								
4	Adaptación de la tensión para ser leída por el ADC	Generador de señales, fuente de alimentación, osciloscopio, polímetro.	PSIM								
5	Regulador PI	PC	CCS								
6	Pruebas	Fuentes,	CCS								

	Experimentales	Osciloscopio, PC									
7	Memoria	PC	Microsoft Word								

CONCLUSIÓN Y TRABAJOS FUTUROS

La intención con la que se desarrolló el presente trabajo era desarrollar un sistema basado en el convertidor inversor de puente completo implementando un sistema de control mediante un micro controlador para que además sirviese con fines didácticos a alumnos de asignaturas relacionadas con la Electrónica de Potencia. Por tanto, tras llevar a cabo un estudio teórico, el desarrollo de un sistema de control, el montaje y puesta en marcha para dos modulaciones diferentes y el desarrollo del software se llega a las siguientes conclusiones:

1) La dificultad principal del trabajo no reside en el montaje del convertidor inversor en sí, ya que gracias al puente completo de transistores solo era necesario generar las señales de gobierno de los transistores y filtrar la salida del convertidor. La parte más compleja llegaba a la hora de acondicionar la señal de salida del convertidor para poder ser leída a través del ADC del micro, y sobre todo a la hora de generar un algoritmo capaz de hallar el valor de la amplitud de la tensión de salida y poder de esta manera implementar el sistema de control. Por otro lado fue complicado hallar un algoritmo que aproximase la generación de la señal PWM Bipolar.

2) Además, como se ha recalcado a lo largo de toda la memoria, se llega a la conclusión de que la modulación PWM Bipolar es mucho más eficaz que la modulación por desplazamiento de fase, gracias a que en este tipo de modulación los armónicos aparecen a frecuencias superiores a 1000 Hz por lo que su filtrado resulta mucho más sencillo y la distorsión armónica es baja.

3) Se demostró que es sencillo interactuar con el convertidor inversor variando determinados parámetros gracias a la interfaz intuitiva de manera que su uso para alumnos en prácticas es recomendable.

4) Se demostró además que el sistema de control implementado, a pesar de su sencillez es totalmente válido ya que se consigue una respuesta estable y se alcanzan resultados válidos a pesar de que la precisión de lectura no es del todo estable al tratarse de una tensión alterna y en el caso de la modulación bipolar, debido a la presencia de ruido por alta frecuencia de conmutación.

Por último se presentan una serie de trabajos futuros para los cuales, el presente proyecto podría ser perfectamente implementado:

1) Podría ser utilizado como etapa de adaptación de una tensión continua obtenida a través de paneles fotovoltaicos.

2) Se podría configurar una interfaz Web a través del puerto Ethernet de la Tiva y conseguir de esta forma interactuar de una forma más sencilla e incluso inalámbrica.

3) Se podría utilizar junto con un convertidor CA/CC para adaptar la tensión generada por un aerogenerador.

BIBLIOGRAFÍA:

- [1] Universidad Carlos III de Madrid. (2013). Plan de prevención de riesgos laborales en la Universidad Carlos III de Madrid.
- [2] Tiva C Series TM4C1294XL Datasheet.
- [3] TivaWare Peripheral Driver Library User's Guide.
- [4] Teoría sobre convertidores inversores utilizada en la asignatura Electrónica de Potencia. Universidad Carlos III de Madrid. Grado de ingeniería en Tecnologías Industriales.
- [5] AD620 Datasheet.
- [6] Narasimha Prasad Prabhu. (2013). *Project Wiki*. Recuperado el 5 de Julio de 2016, de <https://sites.google.com/site/narasimhaweb/projects/phase-shifted-pwm-on-tiva-c-series-microcontrollers-with-dead-time-control>
- [7] Desarrollo de sistema de control de convertidores de potencia CC/CC Dual Active Bridge para plataforma para prácticas de electrónica de potencia. Gonzalo Barrios Gámez (2016). Universidad Carlos III de Madrid. Escuela Politécnica Superior.

A. CÓDIGO DE PROGRAMA PARA MODULACIÓN POR DESPLAZAMIENTO DE FASE:

```

//*****
// Universidad Carlos III de Madrid
// Escuela politécnica superior de Leganés
// Departamento de tecnología electrónica
//
// Trabajo fin de Grado. Desarrollo de sistema de control de convertidores de
// potencia conmutados CC/CA.
// Víctor Arnáiz García-Mora
// Tutor: Pablo Zumel Vaquero
//
// Madrid, Septiembre 2017
//*****

#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_ints.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/debug.h"
#include "driverlib/fpu.h"
#include "driverlib/gpio.h"
#include "driverlib/interrupt.h"
#include "driverlib/pin_map.h"
#include "driverlib/rom.h"
#include "driverlib/rom_map.h"
#include "driverlib/sysctl.h"
#include "driverlib/timer.h"
#include "driverlib/uart.h"
#include "utils/uartstdio.h"
#include "driverlib/pwm.h"
#include "driverlib/adc.h"
#include <math.h>
//*****
//VARIABLES GLOBALES DEL PROGRAMA:

uint32_t g_ui32SysClock;
uint32_t Vo[1];
int_fast32_t int32IntegerPartT;
int_fast32_t int32FractionPartT;
int_fast32_t int32IntegerPartD;
int_fast32_t int32FractionPartD;

float fVoant=0.0; //Almacena el valor máximo de tensión leída a través del ADC
float fVo=0.0; //Almacena cada medida realizada por el ADC en una variable
float
float fVomed[100]; //Almacena 100 máximos de tensión para realizar una media
aritmética
float tension; // Almacena la suma de 100 tensiones
float tensionfinal; // Amplitud de la tensión de salida del convertidor
inversor
float tensionref=0; // Tensión de referencia a alcanzar

```

```

float sumatensiones=0; // Almacena la suma de 100 tensiones
float d=0.3; // Ciclo de trabajo
char seleccion2,selec; //Selección lazo cerrado // selección menú principal
char selecdeadb; // Selección en menú deadband
char selecdef=3; // Selección desfase lazo abierto
float errorahora=0; //Error instantaneo
float errorganancia=0; // Error* ganancia
float K=0.0002; // Ganancia del lazo cerrado
char modo_control=2; // Modo_control=1 -> Lazo cerrado, modo_control=0 ->
Lazo Abierto
float dantes=0; // Ciclo de trabajo anterior para lazo cerrado
int deadband=6550; //Tiempo muerto
float offset=70.46; // Offset introducido por el AD620
int count=0; // Para contar hasta 0.02 (periodo de la tensión de salida del
puente inversor)
int j=0; // Para contar hasta 100 y hacer la media de 100 valores

//FUNCIONES DEL PROGRAMA:

//Función que muestra el menú con los diferentes desfases en Lazo Abierto
void MenuDesfase(){

    UARTprintf("\n Seleccione un desfase:\n\n 1. Desfase 10 por ciento \n
2. Desfase 20 por ciento \n 3. Desfase 30 por ciento \n "
"4. Desfase 40 por ciento \n 5. Desfase 50 por ciento \n 6. Desfase 60
por ciento \n 7. Desfase 70 por ciento \n 8. Desfase 80 por ciento \n 9. Desfase
90 por ciento \n");

}

//Función que muestra las diferentes tensiones de referencia en Lazo Cerrado
void MenuTension(){
    UARTprintf("\nSeleccione la tension de salida que desea alcanzar:\n\n
1. 18 V \n 2. 19 V \n 3. 20 V \n 4. 21 V \n 5. 22 V \n 6. 23 V \n 7 .24 V
\n\n ");

}

//Función para modificar el tiempo muerto
void ModificarDeadBand(char seleccion){
    if(seleccion=='1'){
        deadband=13100;

    } else if(seleccion=='2'){
        deadband=19650;

    } else if(seleccion=='3'){
        deadband=26200;

    } else if(seleccion=='4') {
        deadband=32750;
    }

}

PWMDeadBandEnable(PWM0_BASE,PWM_GEN_0,deadband,deadband);

PWMDeadBandEnable(PWM0_BASE,PWM_GEN_1,deadband,deadband);
}

//Función que da el valor de tensión de referencia escogido en Lazo Cerrado
void LazoCerrado(char seleccion2){
    if(seleccion2=='1'){ tensionref=18;

```



```

        } else if(seleccion2=='2'){
            tensionref=19;
        } else if(seleccion2=='3'){
            tensionref=20;
        } else if(seleccion2=='4') {
            tensionref=21;
        } else if(seleccion2=='5'){
            tensionref=22;
        } else if(seleccion2=='6'){
            tensionref=23;
        } else if( seleccion2=='7'){
            tensionref=24;
        }
    }
}
//Función que da el valor de desfase escogido al ciclo de trabajo en Lazo
Abierto
void LazoAbierto(char seleccion){
    if(seleccion=='1'){
        d=0.1;

    } else if(seleccion=='2'){
        d=0.2;

    } else if(seleccion=='3'){
        d=0.3;

    } else if(seleccion=='4') {
        d=0.4;

    } else if(seleccion=='5'){
        d=0.5;

    } else if(seleccion=='6'){

        d=0.6;
    } else if(seleccion=='7') {

        d=0.7;
    } else if(seleccion=='8'){
        d=0.8;
    } else if(seleccion=='9'){
        d=0.9;
    }
}

    PWMPeriodSet(PWM0_BASE,PWM_OUT_2,PWMGenPeriodGet(PWM0_BASE,
PWM_GEN_1)*d); //Asigna el valor de ciclo de trabajo escogido al generador de
PWM
}

//Función que muestra las opciones del menú principal, además de gestionar el
resto de funciones en función de la elección.
void MenuPrincipal(){
    UARTprintf("\n BIENVENIDO AL SISTEMA \n\n 1.Lazo Abierto \n
2.Modificar Deadband \n 3. Ver propiedades actuales del convertidor \n 4.
Lazo Cerrado \n\n");

```

```

selec = UARTCharGet(UART0_BASE); //Almacena la elección

if(selec=='1'){ modo_control='0'; //Lazo Abierto
    MenuDesfase(); //Muestra los diferentes desfases a elegir
    selecdesf = UARTCharGet(UART0_BASE); //Almacena la elección

    } else if(selec=='2'){
        UARTprintf("\n MODIFICAR DEADBAND \n\n 1.10 Por ciento \n 2.15
Por ciento \n 3. 20 Por ciento \n 4. 25 Por ciento \n\n"); //Muestra los
diferentes tiempos muertos
        selecdadb = UARTCharGet(UART0_BASE); //Almacena la elección
        ModificarDeadBand(selecdadb); //Recibe la elección como
parámetro para modificar el deadband

    } else if(selec=='3'){
        UARTprintf("Desfase = %3d.%03d\n", int32IntegerPartD,
int32FractionPartD); // Muestra el desfase en ese momento
        UARTprintf("Tensión = %3d.%03d\n", int32IntegerPartT,
int32FractionPartT); // Muestra la Amplitud de la tensión de salida del
inversor

    } else if(selec=='4'){

        MenuTension(); //Muestra las tensiones de referencia
        seleccion2=UARTCharGet(UART0_BASE); //Almacena elección
        modo_control='1'; //Lazo cerrado

    } else {
        UARTprintf("\n El comando introducido no es válido, por favor,
inténtelo de nuevo.");
    }
}

//*****

#ifdef DEBUG
void
__error__(char *pcFilename, uint32_t ui32Line)
{
}
#endif

//*****
// Rutina de atención a la interrupción del Timer0 channel A
//*****
void
Timer0IntHandler(void)
{
    ADCProcessorTrigger(ADC0_BASE, 3); //Genera un trigger para indicar
comienzo de conversión
    while(!ADCIntStatus(ADC0_BASE, 3, false)) //Espera a que la conversión
finalice
    {
    }
}

```

```

    ADCIntClear(ADC0_BASE, 3); //Limpia flag de interrupción del adc
    ADCSequenceDataGet(ADC0_BASE, 3, Vo); // Leer el valor del ADC.

    fVo=Vo[0]; //Almaceno Vo en una variable tipo float
    if(fVo>fVoant){ //
        fVoant=fVo; // Algoritmo para almacenar valor máximo de
tensión (Amplitud)
    } else {fVoant=fVoant;}

    if(count>=200){ // 200*(Periodo del timer)=0.02s

        fVomed[j]=fVoant; //Almaceno el máximo de un periodo (0.02s)
        sumatensiones=sumatensiones+fVomed[j]; //Acumula la suma de 100 máximos
        j++; //Para cambiar a la posición siguiente hasta acumular 100 valores
        count=0; //Para contar un nuevo periodo
        fVoant=0; //Para almacenar un máximo del siguiente periodo
        if(j==100){ //
            tension=sumatensiones; //
            sumatensiones=0; // Algoritmo para hacer cambio de unidad
            tensionfinal=tension/100; //
            tensionfinal=tensionfinal*3.3; //
            tensionfinal=tensionfinal/4096; //
            tensionfinal=tensionfinal*36.85; //
            tensionfinal=tensionfinal-offset; //
            j=0;}
        LazoCerrado(seleccion2); // Selecciona tensión de referencia
        if(modos_control=='1'){ // Lazo cerrado
            errorahora=tensionfinal-tensionref; //
            errorganancia=errorahora*K; // Regulador PI
            dantes=d; //
            d=errorganancia+dantes; //

            PWMPulseWidthSet(PWM0_BASE, PWM_OUT_2, PWMGenPeriodGet(PWM0_BASE,
PWM_GEN_1)*d); // Aplica al generador PWM el nuevo valor
        } else { LazoAbierto(selecdesf); } // Lazo Abierto

    }
    count++;

    //Algoritmo para mostrar números decimales
    int32IntegerPartT=(int32_t)tensionfinal;
    int32FractionPartT=(int32_t)(tensionfinal*1000.0f);
    int32FractionPartT=int32FractionPartT-(int32IntegerPartT*1000);
    if(int32FractionPartT<0){
        int32FractionPartT*= -1;}
    //Algoritmo para mostrar números decimales
    int32IntegerPartD=(int32_t)d;
    int32FractionPartD=(int32_t)(d*1000.0f);
    int32FractionPartD=int32FractionPartD-(int32IntegerPartD*1000);
    if(int32FractionPartD<0){
        int32FractionPartD*= -1;}

```

```

// Limpia la interrupción del timer0 channel A

TimerIntClear(TIMER0_BASE, TIMER_TIMA_TIMEOUT);

}

void ConfigureUART(void)
{
//Se habilitan los periféricos de propósito general que utilizan la UART.
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
//Se habilita la UART0.
SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
//Se configuran los pines de propósito general que se van a utilizar en modo
UART.
GPIOPinConfigure(GPIO_PA0_U0RX);
GPIOPinConfigure(GPIO_PA1_U0TX);
GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);
//Se habilitan las interrupciones que se generan cuando se termina de enviar
o recibir por el puerto serie.
UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT);
//Se habilita la UART y se configura la velocidad a la que transmitirá y
recibirá datos, en este caso será de 115200 baudios/segundo.
UARTStdioConfig(0, 115200, g_ui32SysClock);
}
//*****

int main(void)
{
// Configura el reloj a 2.86MHz para poder ajustar a 50 Hz las señales PWM
g_ui32SysClock = SysCtlClockFreqSet((SYSCTL_XTAL_25MHZ |
SYSCTL_OSC_MAIN |
SYSCTL_USE_PLL |
SYSCTL_CFG_VCO_320), 2860000);

// PWM0.....

SysCtlPWMClockSet(SYSCTL_PWMDIV_1); //Sincronización PWM con el reloj
SysCtlPeripheralEnable(SYSCTL_PERIPH_PWM0); //Habilito periférico pwm
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF); //Habilito periféricos F donde
se genera la salida pwm
GPIOPinConfigure(GPIO_PF0_M0PWM0); //Configuro puertos F como salida de pwm
GPIOPinConfigure(GPIO_PF1_M0PWM1);
GPIOPinTypePWM(GPIO_PORTF_BASE,GPIO_PIN_0); //Configuro el puerto en modo pwm
GPIOPinTypePWM(GPIO_PORTF_BASE,GPIO_PIN_1);

PWMGenConfigure(PWM0_BASE,PWM_GEN_0,PWM_GEN_MODE_UP_DOWN|PWM_GEN_MODE_NO_SYNC
); //Configuro el generador en modo UP/DOWN
PWMGenPeriodSet(PWM0_BASE, PWM_GEN_0, 262000); //Asigno el periodo en
número de ciclos como el periodo de conmutación
PWMPulseWidthSet(PWM0_BASE,PWM_OUT_0,PWMGenPeriodGet(PWM0_BASE,
PWM_GEN_0)*50/100);//Ciclo de trabajo 50%
PWMDeadBandEnable(PWM0_BASE,PWM_GEN_0,deadband,deadband); //Tiempo muerto
//La señal generada en PF1 es la negada de PF0

```

```

PWMOutputState(PWM0_BASE,PWM_OUT_1_BIT | PWM_OUT_0_BIT, true);
PWMGenEnable(PWM0_BASE, PWM_GEN_0); //Activo el generador

//PWM1.....

SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
GPIOPinConfigure(GPIO_PF2_M0PWM2);
GPIOPinConfigure(GPIO_PF3_M0PWM3);
GPIOPinTypePWM(GPIO_PORTF_BASE,GPIO_PIN_2);
GPIOPinTypePWM(GPIO_PORTF_BASE,GPIO_PIN_3);
PWMGenConfigure(PWM0_BASE,PWM_GEN_1,PWM_GEN_MODE_DOWN|PWM_GEN_MODE_NO_SYNC);
PWMGenPeriodSet(PWM0_BASE, PWM_GEN_1, 131000);
//Configuro el desfase, inicialmente igual a 30% de un semiperiodo
PWMPulsewidthSet(PWM0_BASE,PWM_OUT_2,PWMGenPeriodGet(PWM0_BASE,
PWM_GEN_1)*30/100);
PWMDeadBandEnable(PWM0_BASE,PWM_GEN_1,deadband,deadband);
PWMOutputState(PWM0_BASE,PWM_OUT_3_BIT | PWM_OUT_2_BIT, true);
HWREG(0x40028000 + 0x0A0)=0x440;
PWMGenEnable(PWM0_BASE, PWM_GEN_1);

// ADC.....

SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0); //Sincronizo ADC con el reloj del
sistema
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE); //Habilito el pin de lectura
GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_3); //Selecciono el pin
correspondiente
ADCSequenceConfigure(ADC0_BASE, 3, ADC_TRIGGER_TIMER, 0); //Genera una lectura
en cada interrupción del timer
ADCSequenceStepConfigure(ADC0_BASE, 3, 0, ADC_CTL_CH0 | ADC_CTL_IE
|ADC_CTL_END); //Selecciona la secuencia 3 de lectura
ADCSequenceEnable(ADC0_BASE, 3);
ADCIntClear(ADC0_BASE, 3); //Limpia la interrupción del timer

// TIMER.....

SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0); //Sincroniza el timer con el
reloj del sistema
TimerConfigure(TIMER0_BASE, TIMER_CFG_PERIODIC); //Se configura como un timer
periódico
TimerLoadSet(TIMER0_BASE, TIMER_A, g_ui32SysClock/10000); //Se configura con
un valor de 100 us
TimerControlTrigger(TIMER0_BASE,TIMER_A,true); //Una lectura del ADC en cada
interrupción

// INTERRUPTIONES.....

IntEnable(INT_TIMER0A); //Habilita interrupción del timer
//Habilitan las interrupciones del timer y la cuenta
TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT);
IntMasterEnable();

```

```

TimerEnable(TIMER0_BASE, TIMER_A);

ConfigureUART();
// Bucle infinito
while(1)
{
    MenuPrincipal();}
}

```

B. CÓDIGO DE PROGRAMA PARA MODULACIÓN PWM BIPOLAR.

```

//*****
// Universidad Carlos III de Madrid
// Escuela politécnica superior de Leganés
// Departamento de tecnología electrónica
//
// Trabajo fin de Grado. Desarrollo de sistema de control de convertidores de
// potencia conmutados CC/CA.
// Víctor Arnáiz García-Mora
// Tutor: Pablo Zumel Vaquero
//
// Madrid, Septiembre 2017
//*****

#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_ints.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/debug.h"
#include "driverlib/fpu.h"
#include "driverlib/gpio.h"
#include "driverlib/interrupt.h"
#include "driverlib/pin_map.h"
#include "driverlib/rom.h"
#include "driverlib/rom_map.h"
#include "driverlib/sysctl.h"
#include "driverlib/timer.h"
#include "driverlib/uart.h"
#include "utils/uartstdio.h"
#include "driverlib/pwm.h"
#include "driverlib/adc.h"
#include <math.h>
#define PI 3.1415

//Variables globales del sistema:

uint32_t g_ui32SysClock;

```

```

uint32_t Vo[1];
int_fast32_t int32IntegerPartT;
int_fast32_t int32FractionPartT;
int_fast32_t int32IntegerPartA;
int_fast32_t int32FractionPartA;

float fVo=0.0; //Almacena el valor de cada medida del ADC
char seleccionprinc=0; //Variable de selección del menú principal
char selecamp; //Variable para seleccionar modulación en amplitud
char selecref; //Variable para seleccionar tensión de referencia
float duty=0.9; //Ciclo de trabajo del generador PWM
float A=1; //Modulación en amplitud
float offset=0; //Almacena la suma de 100 medidas del ADC y las promedia para
calcular el offset
float offsetfin; //Cuadrado de la variable offset
int count; //Cuenta hasta 20, equivalente a un ciclo de la tensión de salida
(0.02s)
int i=0; //Variable para recorrer el vector que da valor al ciclo de trabajo
(sen[20])
float tensionfinal; // Amplitud de la tensión de salida del convertidor
float Aantes; //Modulación en amplitud en el ciclo anterior
float errorahora=0; //Error instantáneo
float errorganancia=0; //Errorahora*ganancia
float K=0.0001; //Ganancia del sistema en lazo cerrado
char modo_control=0; // Modo_control=1 -> Lazo cerrado, modo_control=0 -> Lazo
Abierto
float tensionref=0; //Tensión de referencia en lazo cerrado
float tensionef=0; // Almacena el cuadrado de 100 muestras y las promedia para
calcular el cuadrado de la tensión eficaz
float tensioneffin=0; // Valor de la tensión eficaz de la señal a la salida
del convertidor
//Variable que almacena el valor del ciclo de trabajo
float
sen[20]={0.5,0.655,0.794,0.905,0.976,1.000,0.976,0.905,0.794,0.655,0.500,0.34
5,0.206,0.095,0.024,0.000,0.024,0.095,0.206,0.345};

//Función en la que se da valor a la tensión de referencia
void LazoCerrado(char seleccion2){
    if(seleccion2=='1'){ tensionref=4;

        } else if(seleccion2=='2'){
            tensionref=6;
        } else if(seleccion2=='3'){
            tensionref=8;
        } else if(seleccion2=='4') {
            tensionref=10;
        } else if(seleccion2=='5') {
            tensionref=12;
        } else if(seleccion2=='6') {
            tensionref=14;
        } else if(seleccion2=='7') {
            tensionref=16;
        }
    }

}

//Función en la que se da valor a la modulación en amplitud
void ModificarAmplitud(char seleccion){

```

```

        if(seleccion=='0'){
            A=1;} else if(seleccion=='1'){
                A=0.9;

            } else if(seleccion=='2'){
                A=0.8;

            } else if(seleccion=='3'){
                A=0.7;

            } else if(seleccion=='4') {
                A=0.6;

            } else if(seleccion=='5'){
                A=0.5;

            } else if(seleccion=='6'){
                A=0.4;

            } else if(seleccion=='7'){
                A=0.3;

            } else if(seleccion=='8'){
                A=0.2;

            } else if(seleccion=='9'){
                A=0.1;
            }

    }
    //Función que muestra las diferentes tensiones de referencia que podemos
    escoger
    void MenuTension(){
        UARTprintf("\n Seleccione la tensión de salida que desea alcanzar:\n\n
        1. 4 V \n 2. 6 V \n 3. 8 V \n 4. 10 V \n 5. 12 V"
        " \n 6. 14 V \n 7. 16 V \n\n");

    }
    //Función que muestra los diferentes valores de modulación en amplitud
    void MenuAmplitud(){

        UARTprintf("\n Seleccione la modulación en amplitud deseada:\n\n 0. 1
        \n 1. 0.9 \n 2. 0.8 \n 3. 0.7 \n"
        " 4. 0.6 \n 5. 0.5 \n 6. 0.4 \n 7. 0.3 \n 8. 0.2 \n 9. 0.1
        \n\n");

    }
    //Función que muestra el menú principal, gestiona los diferentes modos de
    control y muestra información
    //su funcionamiento es idéntico al explicado en el código de modulación por
    desplazamiento de fase.
    void MenuPrincipalPWM(){
        UARTprintf("\n BIENVENIDO AL SISTEMA \n
        ..... \n\n 1.Lazo Abierto \n 2. Ver amplitud de
        tension y desfase \n 3. Lazo Cerrado \n\n");

        seleccionprinc = UARTCharGet(UART0_BASE);
    }

```



```

        if(seleccionprinc=='1'){
            modo_control='0';
            MenuAmplitud();
            selecamp = UARTCharGet(UART0_BASE);

        } else if(seleccionprinc=='2'){
UARTprintf("Amplitud de la tensión de salida = %3d.%03d\n",
int32IntegerPartT, int32FractionPartT);
UARTprintf("Modulacion en amplitud = %3d.%03d\n", int32IntegerPartA,
int32FractionPartA);
        } else if(seleccionprinc=='3'){
            MenuTension();
            selecref = UARTCharGet(UART0_BASE);
            modo_control='1';
        }
    }

//*****

#ifdef DEBUG
void
__error__(char *pcFilename, uint32_t ui32Line)
{
}
#endif

//*****
// Rutina de atención a la interrupción del Timer0 channel A
//*****
void
Timer0IntHandler(void)
{
    ADCProcessorTrigger(ADC0_BASE, 3); // Genera un trigger para indicar
    comienzo de conversión
    while(!ADCIntStatus(ADC0_BASE, 3, false)) // Espera a que la conversión
    finalice
    {
    }

    ADCIntClear(ADC0_BASE, 3); // Limpia flag de interrupción del adc
    ADCSequenceDataGet(ADC0_BASE, 3, Vo); // Leer el valor del ADC.
    // Almaceno Vo en una variable tipo float
    fVo=Vo[0];

    if((fVo<3000)&(fVo>1500)){ // Limita entre valores de tensión máximo y mínimo
    para evitar la influencia de ruido
        offset=offset+fVo; // Acumula la suma en cada muestra
        tensef=tensef+(fVo*fVo); // Acumula la suma de los cuadrados de cada
    muestra
        count++;}

    if(count>99){

```

```

    tensionef/=100;//Promedio de la suma del cuadrado de 100 muestras

    offset/=100; //Valor del offset
    offsetfin=offset*offset; //Cuadrado de la variable offset

    tensioneffin=tensionef-offsetfin;
    tensionfinal= sqrt(tensioneffin); //Valor de la tensión eficaz
    tensionfinal=tensionfinal/0.707; //Valor de amplitud de la tensión de
salida del convertidor inversor
    tensionfinal=tensionfinal*3.3; //
    tensionfinal=tensionfinal/4096; // Conversión a unidades de tensión
    tensionfinal=tensionfinal*38; //
    count=0;
        offset=0;
        tensionef=0;
}

//Generación del ciclo de trabajo para modulación bipolar
duty=(A*(sen[i]-0.5)+0.5);
if(i>=19){i=0;} else { i++;} // Para recorrer el vector sen[20] con los
valores del ciclo de trabajo
//Limita el ciclo de trabajo para que no sea mayor que 0.99 ni menor que
0.03 por problemas con el deadband
if(duty>=(0.99)){duty=(0.99);} else if(duty<=(0.03)){duty=(0.03);}
//Asigna el valor del ciclo de trabajo en cada interrupción
PWMPulseWidthSet(PWM0_BASE,PWM_OUT_0,PWMGenPeriodGet(PWM0_BASE,
PWM_GEN_0)*duty);

LazoCerrado(seleceref); //Asigna la tensión de referencia escogida en el menú
if(modo_control=='1'){
    errorahora=tensionref-tensionfinal; //
    errorganancia=errorahora*K; // LAZO CERRADO
    Aantes=A; //
    A=errorganancia+Aantes; //

} else {ModificarAmplitud(selecamp);} // LAZO ABIERTO

//Algoritmo para mostrar números decimales
int32IntegerPartT=(int32_t)tensionfinal;
int32FractionPartT=(int32_t)(tensionfinal*1000.0f);
int32FractionPartT=int32FractionPartT-(int32IntegerPartT*1000);
if(int32FractionPartT<0){
    int32FractionPartT*= -1;}
//Algoritmo para mostrar números decimales
int32IntegerPartA=(int32_t)A;
int32FractionPartA=(int32_t)(A*1000.0f);
int32FractionPartA=int32FractionPartA-(int32IntegerPartA*1000);
if(int32FractionPartA<0){
    int32FractionPartA*= -1;}

// Limpia la interrupción del timer0 channel A
TimerIntClear(TIMER0_BASE, TIMER_TIMA_TIMEOUT);

}

```

```

//Misma configuración de la UART que en la programación para desplazamiento
de fase
void
    ConfigureUART(void)
    {

        SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
        SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
        GPIOPinConfigure(GPIO_PA0_U0RX);
        GPIOPinConfigure(GPIO_PA1_U0TX);
        GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);
        UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT);
        UARTStdioConfig(0, 115200, g_ui32SysClock);
    }

//*****

int
main(void)
{

    // Se configura el reloj del sistema a 120 MHz
    g_ui32SysClock = MAP_SysCtlClockFreqSet((SYSCTL_XTAL_25MHZ |
                                                SYSCTL_OSC_MAIN |
                                                SYSCTL_USE_PLL |
                                                SYSCTL_CFG_VCO_480), 120000000);

    // PWM0.....
    // En este caso solo hay un generador PWM, las funciones empleadas son
    iguales que en el modo desplazamiento de fase
    SysCtlPWMClockSet(SYSCTL_PWMDIV_1);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_PWM0);

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
    GPIOPinConfigure(GPIO_PF0_M0PWM0);
    GPIOPinConfigure(GPIO_PF1_M0PWM1);
    GPIOPinTypePWM(GPIO_PORTF_BASE, GPIO_PIN_0);
    GPIOPinTypePWM(GPIO_PORTF_BASE, GPIO_PIN_1);

    PWMGenConfigure(PWM0_BASE, PWM_GEN_0, PWM_GEN_MODE_UP_DOWN | PWM_GEN_MODE_NO_SYNC
    );
    PWMGenPeriodSet(PWM0_BASE, PWM_GEN_0, 12000);
    PWMPulseWidthSet(PWM0_BASE, PWM_OUT_0, PWMGenPeriodGet(PWM0_BASE,
    PWM_GEN_0)*(0.5));
    PWMDeadBandEnable(PWM0_BASE, PWM_GEN_0, 600, 600);
    PWMOutputState(PWM0_BASE, PWM_OUT_1_BIT | PWM_OUT_0_BIT, true);
    PWMGenEnable(PWM0_BASE, PWM_GEN_0);

    // ADC.....
    // Igual que en desplazamiento de fase
    SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);

```

```

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
    GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_3);
    ADCSequenceConfigure(ADC0_BASE, 3, ADC_TRIGGER_TIMER, 0);
    ADCSequenceStepConfigure(ADC0_BASE, 3, 0, ADC_CTL_CH0 | ADC_CTL_IE
|ADC_CTL_END);
    ADCSequenceEnable(ADC0_BASE, 3);
    ADCIntClear(ADC0_BASE, 3);


    // TIMER.....
    // Igual que en desplazamiento de fase pero en este caso el periodo es
igual a 1 ms

    SysCtlPeripheralEnable(SYSCTL_PERIPH_TIMER0);
    TimerConfigure(TIMER0_BASE, TIMER_CFG_PERIODIC);
    TimerLoadSet(TIMER0_BASE, TIMER_A, g_ui32SysClock/1000);
    TimerControlTrigger(TIMER0_BASE, TIMER_A, true);


    // Igual que en desplazamiento de fase
    IntEnable(INT_TIMER0A);

    TimerIntEnable(TIMER0_BASE, TIMER_TIMA_TIMEOUT);

    IntMasterEnable();
    TimerEnable(TIMER0_BASE, TIMER_A);


    ConfigureUART();

    // Bucle infinito

    while(1)
    {
        MenuPrincipalPWM();
    }
}

```